Improvement of Automatic Target Recognition Through Synthetic Data Augmentation

Ciaran Sanford[1], Oscar Bryan[1], Tom Fincham-Haines[1], and Alan Hunter[1]

[1]University of Bath, Claverton Down, Bath, UK, BA2 7AY

Contact author: Ciaran Sanford, University of Bath, Claverton Down, Bath, UK, BA2 7AY, c.sanford@bath.ac.uk

Abstract: *Data sets of well-labelled and diverse acoustic imagery of the seabed are scarce. However, a recent breakthrough in synthetic aperture sonar (SAS) image simulation has facilitated the rapid generation of realistic echo data. The synthetic data include important aspects of the acoustic wave physics, such as aspect-dependence, layover, diffraction, speckle, focusing errors, and artefacts. Moreover, it provides high-fidelity label information. This combination of speed, realism, and detail has enabled the use of synthetic data to improve the volume and diversity of training data for deep learning algorithms in automatic target recognition (ATR). We present an overview of the rapid simulation model, alongside an existing SAS siulation model, and demonstrate its application to ATR training for the detection and classification of underwater munitions and unexploded ordnance.*

Keywords: *ATR, UXO, Machine learning, SAS, Simulation*

1. INTRODUCTION

Unexploded ordnance (UXO) remediation is an important application of underwater acoustics, due in part to the vast quantities of munitions that were disposed of at sea following the Second World War [1]. Surveillance and cataloguing of these UXO "dumpsites" is possible due to platforms carrying high resolution synthetic aperture sonar (SAS) sensors. However, detection and classification of targets in the dumpsites is a task suited to automated approaches due to the quantity of data [2]. These automated target recognition (ATR) models generally employ supervised machine learning, using labelled images to train model parameters. However, this requires a large library of example data, which is difficult or time consuming to obtain, and can suffer from inaccurate human labelling.

A potential solution is augmentation of datasets through simulation. Simulation enables the creation of highly diverse datasets in a less time consuming manner than collection of real data would allow. In addition, high resolution target labels can be created, ensuring accuracy for supervised learning. Previously, creation of synthetic images has been achieved by rendering images of 3D models, followed by adding noise to approximate speckle [3]. This allows SAS images to be "faked" quickly, with a trade-off between speed and realism. However, a recent breakthrough in SAS simulation has enabled the rapid generation of realistic, coherent echo data [4]. This eases the aforementioned trade-off and allows for the generation of large, realistic datasets for machine learning applications.

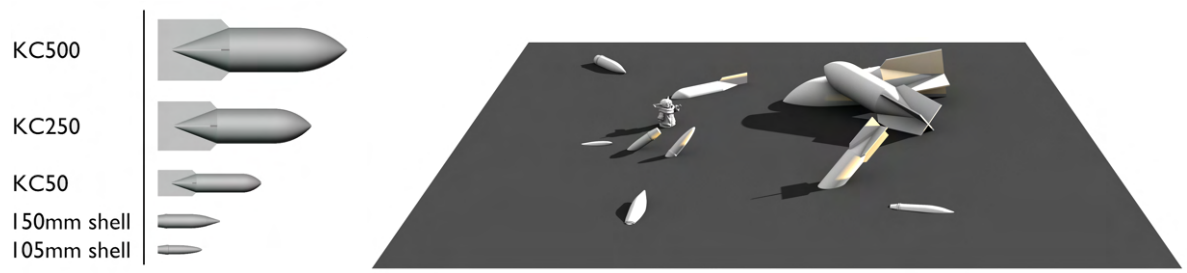
The process outlined in this paper is a first step towards improvement of ATR models through the use of simulated data. We do this by comparison between models trained on data from a "basic" image simulator and the "advanced" raw data simulator. We summarise the data generation and machine learning methods with a comparison between the simulators, before employing the models in three combinations of training and testing. Detection and classification results for each combination are shown, concluding with a discussion of the impact of simulation realism on the ATR models.

2. SIMULATION METHODS

Training of the machine learning models for ATR requires a vast amount of data. A processing chain was developed to generate and simulate a large number of data products for a diverse set of scenes containing common target UXOs [5]. The data products are generated using a combination of software. For scene generation, we use the open-source modelling software Blender [6]. Simulation of raw data, images, and labelled segmentation maps is then performed in MATLAB.

2.1. SCENE GENERATION

Scene generation comprises two steps: target selection and target placement. These are common to both simulators and are performed in Blender. Targets are uniformly selected from a list of five example UXOs commonly found at the Skaggerak dumpsite [5], illustrated in figure 1a. The seafloor is represented by a flat 10m by 10m plane. Targets are dropped onto the seafloor and then perturbed a small amount to diversify target orientation and burial state. Each target is assigned an an index, which is later used to create segmentation maps. An example scene is given in figure 1b.



(a) *(b)*

Figure 1: (a) UXOs used as targets in the simulated scenes, selected from example UXOs found at the Skaggerak dumpsite, used in (b) an example scene with 8 targets. [this example should carry through to the demo results later]

2.2. IMAGE SIMULATION (BASIC SIMULATOR)

The sonar transducers are mimicked using an orthographic light source, angled to highlight geometry in a comparable way to a SAS. The scene is rendered from an orthographic camera placed above the scene. Approximation of SAS speckle is achieved via multiplicative noise calculated from floor speckle statistics of real SAS data. A complete description of this process can be found in [5]. Finally, a high resolution segmentation map is created using a second render from the orthographic camera, where each object is encoded with a colour defined by its index.

2.3. RAW DATA SIMULATION (ADVANCED SIMULATOR)

Simulation of raw data utilises the rapid wavefield simulation method documented in [4]. This requires that an orthographic camera render the scene from all angles inside the transducer beamwidth. These renders are used, via projection mapping, to determine the amplitude of the wavefield as a function of aspect angle and frequency. This incorporates geometric effects such as layover and aspect dependence. The coherent component of the wavefield is modelled by incorporating a stochastic phase function in the Fourier domain. The coherent wave spectrum is propagated to the synthetic aperture using Fourier extrapolation, and data is generated by interpolating the resulting spectrum. Finally, the data is back-projected to form images with dimensions matching those generated with the image simulator.

3. MACHINE LEARNING METHODS

The popular YOLO (you only look once) v5s model [7] was selected model and trained on datasets from both simulators, resulting in two sets of trained parameters. The YOLO family of models perform object detection and classification in a single forward pass. These models consist of key components: a backbone, a neck, and a regressor/classifier head. The backbone performs feature extraction by progressively compressing the image through a series of convolutional operations. Intermediate activations from the backbone are passed to the neck which aggregates features at multiple scales and resolutions. Anchors, which are predefined bounding boxes of different sizes and aspect ratios derived from the dataset, are used to generate region proposals at various locations in the image. The aggregated features corresponding to these region proposals are then passed through a regressor/classifier head. For each anchor, the objectness, class label, and bounding box shape and size are refined.

The YOLO v5s model used in this study [7], incorporates the darknet-58 [8] residual convo-

lutional network as its backbone, and the path aggregated network [9] as its neck. The location of the targets was defined by bounding boxes generated from segmentation maps. Synthetic data consisting of 4,000 images were generated from each simulator, with 250 allocated for validation, 1,250 for testing, and 2,500 for training. Both models were trained for 15 epochs with a batch size of 16, facilitating the convergence of the network.

4. RESULTS

4.1. SYNTHETIC DATA COMPARISON

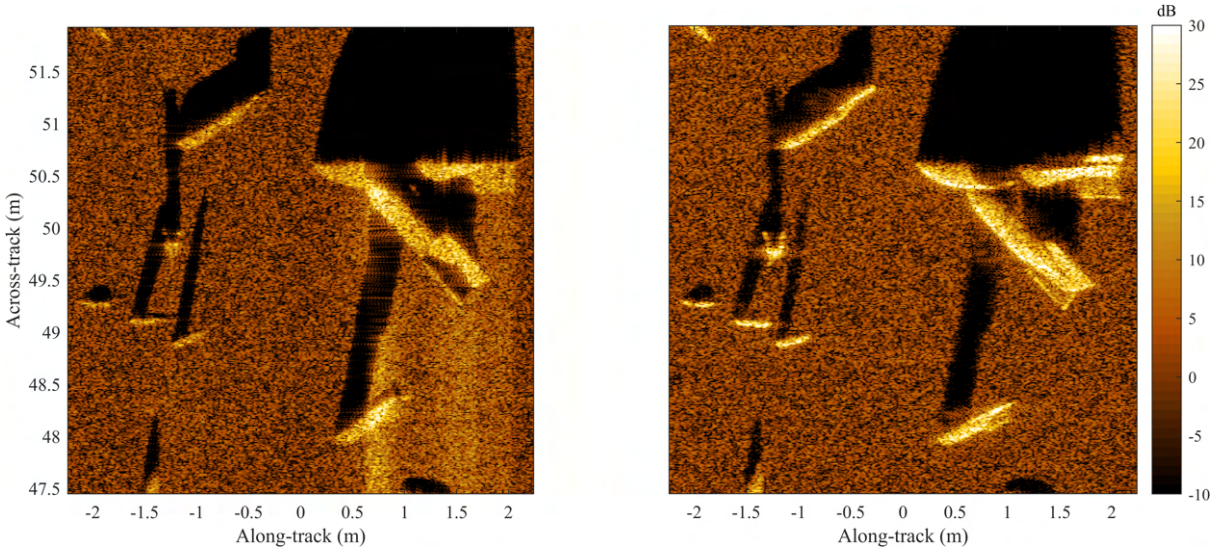


Figure 2: Output image for the example scene in figure 1b from (a) the basic image simulator and (b) the advanced simulator.

Figure 2 shows the output images from each simulator. The image from the advanced simulator (figure 2b) displays geometric effects not present in the basic simulator output (figure 2a) such as layover and aspect-dependent highlights, particularly visible on the large bomb in the upper right of the image. In addition, the target shadows have penumbral regions that are more accurate to real SAS data. Figure 3 is a gallery of random example images (from the advanced simulator).

4.2. MACHINE LEARNING PERFORMANCE COMPARISON

The performance of each model after training was evaluated using three different combinations of training and testing. These were 1) Training on basic data, testing on basic data, 2) Training on advanced data, testing on advanced data, and 3) Training on basic data, testing on advanced data. The implied fourth combination, training on advanced data and testing on basic data, was omitted as it represents an unlikely situation where the training data is more realistic than the data the classifier is employed on. The scenes in the training set are structurally identical across simulators, containing the same targets.

Figure 4 shows data products generated for the example scene from figure 1b. Figure 4a shows a segmentation map, with bounding boxes overlaid for each target. The colours and

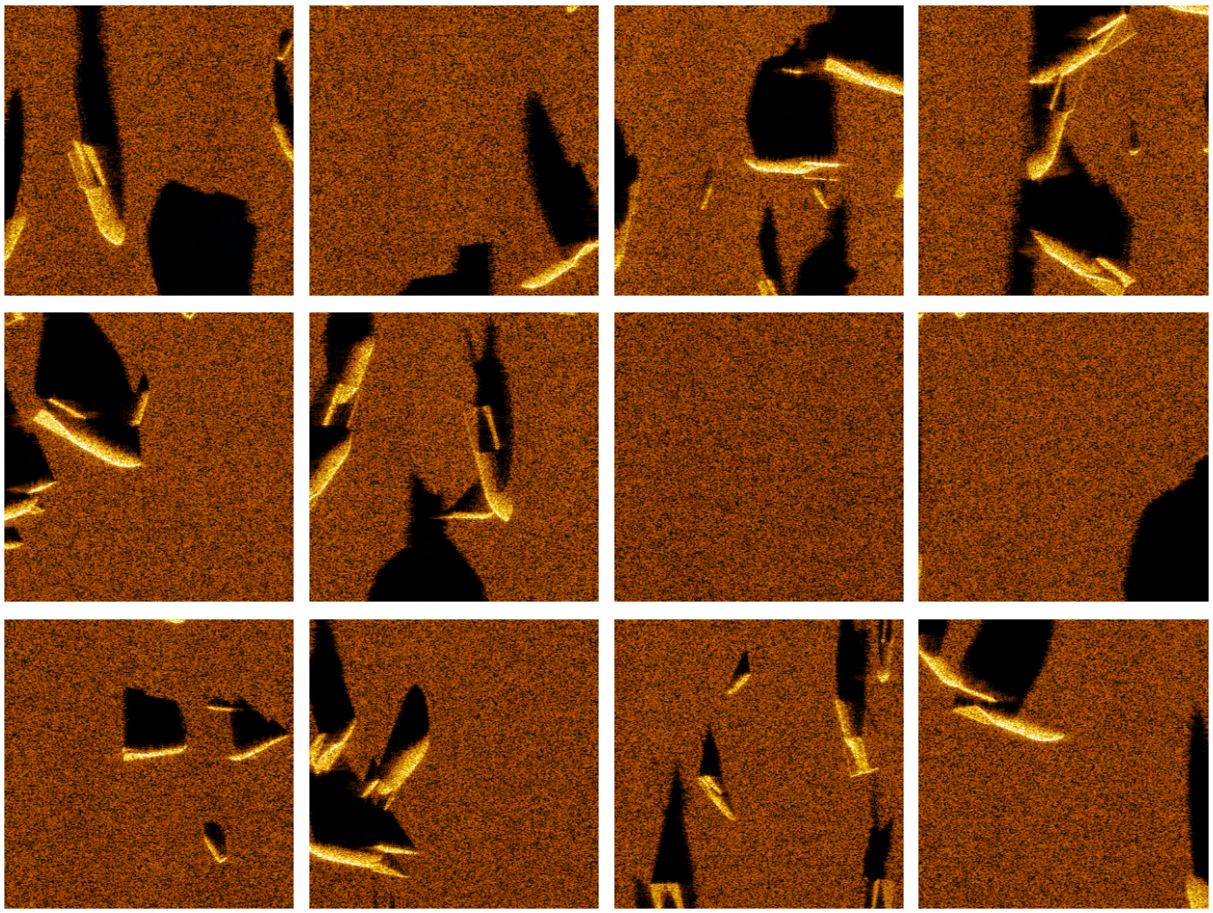


Figure 3: Selection of example scenes from the advanced simulator.

Object name	Class colour	Object index
KC500 Bomb	Green	8
KC250	Yellow	12
KC50	Cyan	3, 11
150mm Shell	Magenta	6, 7, 10
105mm Shell	White	1, 2, 4, 5, 9

Table 1: Segmentation map key for figure 2.

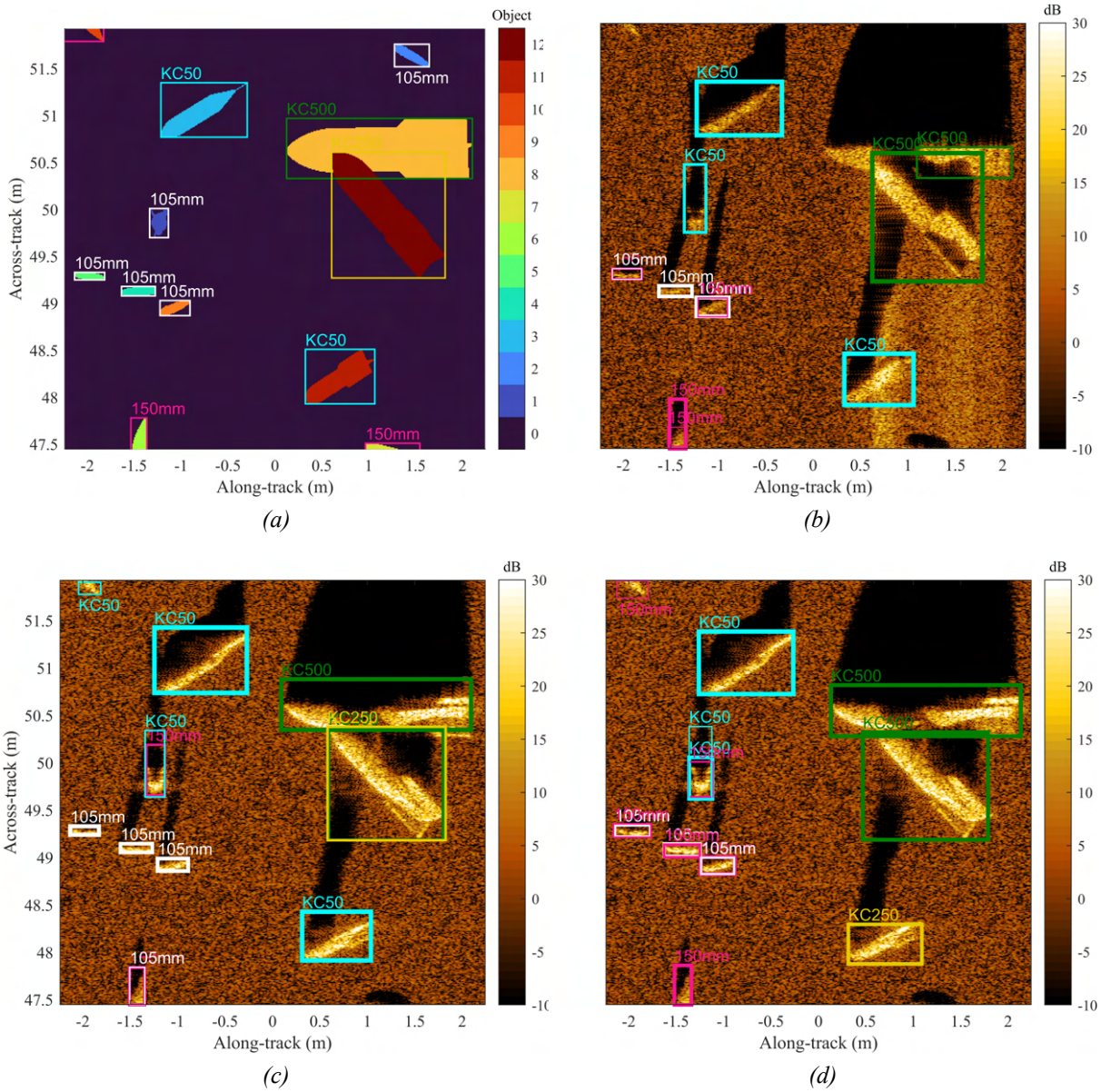


Figure 4: Machine learning models tested on the example scene in figure 1b, with (a) ground truth segmentation map alongside classification results for (b) training on basic, testing on basic, (c) training on advanced, testing on advanced, and (d) training on basic, testing on advanced. Bounding box line thickness represents model confidence.

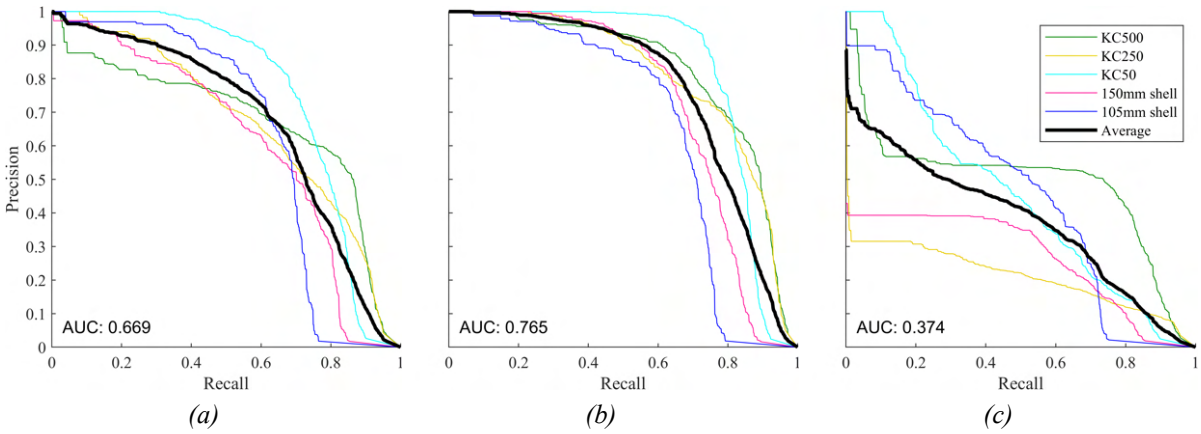


Figure 5: Precision-recall curves per class for (a) training on basic, testing on basic, (b) training on advanced, testing on advanced, and (c) training on basic, testing on advanced.

object indices associated with each class are indicated in table 1. Figures 4b, 4c and 4d show qualitative results of each model. These use a confidence threshold of 0.2 and a thicker bounding box indicates a higher confidence. Differences can be clearly seen between each result, with the second case (train advanced, test advanced) performing significantly better than the others in terms of distinguishing between the larger targets. The first and third cases (both trained on basic data) show some difficulty with smaller targets in particular, with overlapping boxes for each instance. The third case performs more poorly than the others, with less success with larger targets and several overlapping boxes.

Performance is quantified with an intersection over union (IoU) test using the model predictions and the ground truth bounding boxes. Figure 5 shows precision-recall curves per class for each testing combination, with the area under the curve (AUC) values indicated. The second test case performs marginally better than the others, with the poorest performance exhibited by the third case (train basic, test advanced). This is understandable when considering that the advanced data incorporates more geometric features (i.e. specular highlights, layover) than the basic data. This may make distinguishing targets in advanced data more reliable, while hindering a model trained without those features.

5. CONCLUSIONS AND FUTURE WORK

We have successfully trained two classifiers based on two different simulation techniques. The first used a basic SAS image simulator that is fast, but trades realism for that speed. The second simulator, while still faster than traditional raw data simulation methods, is slower but incorporates more realistic features such as specular highlights and layover. The models were separately trained on datasets comprising 2500 images from each simulator using the YOLO v5s model and employed on a further set of 1250 images for testing.

The models show the best results when tested on data from the simulator they were trained on. Significant degradation in prediction quality results when testing the basic model on advanced data, halving the AUC score, as is expected, as the advanced data incorporates more realistic features. We hypothesise that realism in training data does have a significant effect on a classifier's performance, and that a more sophisticated simulation model has a positive effect on performance. We assert that the more closely a simulation model can represent real data (including introducing errors such as multipath), the more effectively the classifier trained on

it can perform. Future work will be focused on testing this hypothesis, by training models on a variety of synthetic data and testing on real labelled data.

6. ACKNOWLEDGEMENTS

This work was supported by the Strategic Environmental Research and Development Program (SERDP) under Grant MR21-1339.

REFERENCES

[1] R. E. Hansen, T. O. Sæbø, O. J. Lorentzen, and S. A. V. Synnes, "Mapping Unexploded Ordnance (UXO) Using Interferometric Synthetic Aperture Sonar," in *Proceedings of the Underwater Acoustics Conference and Exhibition (UACE)*, pp. 687–694.

[2] J. C. Isaacs, "Sonar automatic target recognition for underwater UXO remediation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 1. IEEE, 2015, pp. 134–140.

[3] M. M. Siddiqui, "Statistical inference for rayleigh distributions," *Journal of Research of the National Bureau of Standards, Sec. D*, vol. 68D, 1964.

[4] C. Sanford, B. Thomas, and A. Hunter, "Fourier-Domain Wavefield Rendering for Rapid Simulation of Synthetic Aperture Sonar Data," in communication.

[5] O. Bryan, T. S. F. Haines, A. Hunter, R. E. Hansen, and N. Warakagoda, "Automatic recognition of underwater munitions from multi-view sonar surveys using semi supervised machine learning: a simulation study," in *International Conference on Underwater Acoustics*, vol. 47, 2022, p. 070018.

[6] The Blender Foundation, "Blender," 2022. [Online]. Available: https://www.blender.org/

[7] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv:2004.10934 [cs, eess]*, no. April, 2020. [Online]. Available: http://arxiv.org/abs/2004.10934

[8] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," in *arXiv*, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[9] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.