Towards a Generic Object Detector

Ashvin Pidaparti;

University of Minnesota

Introduction

In the field of Autonomous Robotics, object detection is a key element for a robot to interact with the world. A robot needs to be aware of its surroundings in order to take the appropriate action for those surroundings. Through extensive research, algorithms to detect objects and determine their location in images have been developed, however, these algorithms require a large amount of high quality images of the objects. Several objects don't have significant data on them, making detecting them through conventional methods challenging. We put forward a method of detecting objects without significant data on them via extracting features and matching the presence of those features against known outputs for objects [3]. This method functions by creating binary feature extractors and using the output of several feature extractors to create a vector representation of an object. This vector representation is then matched against a list of several objects that were not in the original dataset and their expected output from those feature extractors to find the closest match. This match is the classification of that object.

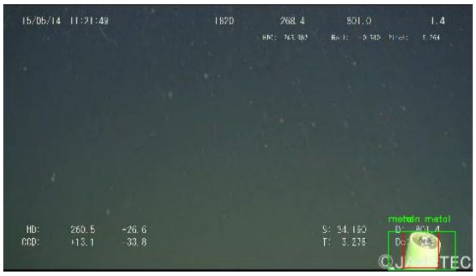

Fig 1. Results of the metal vs non-metal feature extractor. Clearly shown, the object detection is of low quality, leading in poor results.

Methods

Feature extractors can be as simple as whether or not an object is made of biological material, or whether or not an object can tangle the propeller of an underwater robot. The Faster R-CNN [2] neural network was attempted to train to locate objects in an image and classify them as biological or non-biological, able to tangle or not able to tangle, and metal or nonmetal. These models would be applied to images and their output was vectorized. The vector representations of objects in the image would be matched against hard-coded representations of known but unseen objects via the K-Nearest Neighbor algorithm. The closest neighbor in feature space will be treated as the classification. These feature extractors were trained using high performance computing resources via the Minnesota Supercomputing Institute. The TrashCan dataset was used for training [1].

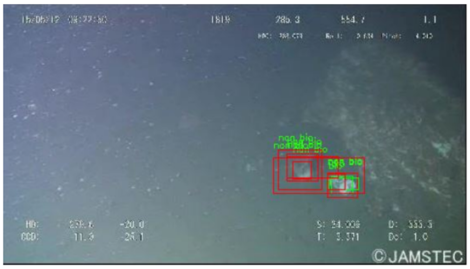

Fig 2. Results of the biological material feature extractor. It is visually apparent that the object detection is not of high quality.

Results

The feature extractors were trained via the Minnesota Supercomputing Institute (MSI). While MSI offers high performance GPUs to assist in training deep neural networks, these GPUs (Tesla K40m and Tesla v100) did not have the required support from PyTorch [4] to fully utilize their capabilities. This imposed a significant time cost to training feature extractors, going up to and exceeding 24 hours for a single epoch, resulting in spurious and inaccurate object detection via feature extractors. The results of the object detection are shown below. As shown, the feature extractors are able to somewhat accurately locate objects in frame, but are not able to accurately classify them nor discard spurious detected objects. Additional training is need to accurately detect objects.

Conclusion

While this implementation of Zero-shot learning has significant flaws relating to object detection, the core method put forward still holds promise. This method is not perfect, as it still requires some human effort to annotate the expected output from feature extractors. However, this still requires significantly less human effort when compared to the requirement for perfectly annotated images.

Acknowledgements

Prof. Junaed Sattar and Jungseok Hong both assisted me greatly in ideating and implementing this work.

References

Hong, Jungseok; Fulton, Michael S; Sattar, Junaed. (2020). TrashCan 1.0 An Instance-Segmentation Labeled Dataset of Trash Observations. Retrieved from the Data Repository for the University of Minnesota, https://doi.org/10.13020/g1gx-y834. Ren, Shaoqing, et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, 2017, pp. 1137–1149., https://doi.org/10.1109/tpami.2016.2577031. C. H. Lampert, H. Nickisch, and S. Harmeling. "Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer". In CVPR, 2009. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32 (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf