# <u>Research Assessment #7</u>

**Date:** 29 October 2021

**Subject:** Computer Vision Object Detection with Improved SSD

**MLA Citation:**

Jin, Liang, and Guodong Liu. "An Approach on Image Processing of Deep Learning Based on Improved SSD." *MDPI*, Multidisciplinary Digital Publishing Institute, 17 Mar. 2021, https://www.mdpi.com/2073-8994/13/3/495/htm.

**Assessment**:

After taking a look at a deformable CNN and YOLO algorithm, the next method for me to tackle was the SSD (Single shot detector) algorithm. The SSD seemed very similar to the YOLO algorithm in that it worked in one stage, unlike the CNN which was a two-stage process. The SSD algorithm also had a similar effectiveness as the YOLO algorithm— with the improvements made by the authors, it was very close in precision and speed. This means that the use case is probably the most important factor in determining which algorithms are right for the job. In this case, the use case was ship sensing through remote sensing images, which was directly related to my research because there were constraints on speed and memory size on the hardware that was used in the study (Liang et. al 1).

An important consideration that this article gave me was the importance of background. Locating an object against a blank background and a busy background are very different difficulties for a computer vision program, since the business of the background increases the number of potential objects that the program has to parse through. As a result, training any model will need to have the target object situated in many different contexts.

A similarity that I saw between the YOLO and SSD algorithms is that they split the region into a grid to work with. As a result, instead of taking one stage for region proposal which, although accurate, is very time consuming, this indiscriminate division can greatly increase the speed of the calculation. In addition, both the SSD and YOLO algorithms had the object confidence score as the main factor, largely because of the nature of the region proposal. Also, since “the selection of region proposal” is not important as long as the bounding boxes are correct, only the most efficient methods that do not compromise in accuracy are the best (Liang et. al 3).

As datasets get incredibly large, as is the case with many remote sensing applications, it takes exponentially longer for algorithms to traverse datasets and make interpretation. The technique that is employed in the SSD algorithm of having a CNN to go deeper and deeper into the features of the image in each layer and identify the objects at lower layers is a unique technique that can be really helpful in processing speed. The part that takes the longest seems to be upsampling and building an infrastructure that connects the upsampling with the downsampling of the CNN. However, compared to other algorithms, this is still fast enough to detect objects in real-time.

I was also able to learn about the feature pyramid network through the course of the article. Since it seemed like a useful tool for connecting the different aspects of the algorithm, I was wondering, how can it be implemented in other algorithms? Which ones might benefit from this and which ones would not? The mean absolute precision continues to seem like the primary means of measuring the effectiveness of an algorithm in conjunction with the runtime, so I will definitely include these in my research paper. In order to learn more about the standard that these algorithms must reach to be effective, I need to know how fast and how compact programs need to be to detect objects in real-time. Throughout my future research, I hope to answer these questions and more.