Research Assessment #4

Date: 23 September 2021

Subject: Machine Learning and Computer Vision

MLA Citation:

G. Castellano, C. Castiello, C. Mencar and G. Vessio, "Crowd Detection in Aerial Images Using Spatial Graphs and Fully-Convolutional Neural Networks," in IEEE Access, vol. 8, pp. 64534-64544, 2020, doi: 10.1109/ACCESS.2020.2984768.

Assessment:

Since I am interested in using computer vision in my original work, I wanted to research how it interacted with deep learning algorithms such as neural networks. The underlying structure in computer vision is often made up of machine learning algorithms, since the computer needs a way to analyze the image that it is looking at. There are multiple techniques for doing this, and although they are very important, the article did not help me learn much in that respect. Instead, I was able to learn about the constraints that can be placed upon a system when there are real world applications.

For example, the article focused on a drone system to analyze crowds and determine a safe place to land. Since the drone needs to be able to fly effectively, the GPU needs to be small and compact to keep it from placing increased drag on the drone. Furthermore, speed is very important in this application because the drone needs to react fast enough to adjust its flight path based on what it sees (Castellano et. al 1). Although these characteristics used to seem universal to me, I now know that this is not the case. If the application had been in a supercomputer that was not going to be

moved, the GPU could be much bigger and could support much more complex algorithms. As a result, size would be less of a constraint.

I also came to the realization that no ML model is going to be perfect. Although one hundred percent is the goal, models are always going to have some level of inaccuracy, or will have to give away precision to increase accuracy. This was the case in the system in the article, which had lower precision as accuracy increased (Castellano et. al 8). However, I found that this is not necessarily a bad thing, since the task at hand may just need the model to classify objects. This makes precision less of a concern. Once again, tradeoffs that can be made are dependent on the application of the ML model.

In terms of algorithmic techniques, processing power is a very real limitation regardless of the application. Especially because I am a student and may not have the access (or money) for extremely high-quality materials, any models that I create would need to be optimized very well. To this end, splitting up input elements into smaller pools could help reduce the strain on the processor. This is because the number of dimensions and analysis required becomes less complex, which lowers runtime considerably.

I was also surprised to realize how much of a factor slight variations in data could impact the success of the model. The fact that this happens means that if I must collect my own data, it needs to be accurate and taken in similar formats to ensure that the model can accurately interpret it. Since this can take up a lot of time, resources, and

may need a controlled environment, trying to use open source training data and repurposing it would seem to be the better option.

At this point, I need to evaluate my own resources to see what type of models I can feasibly implement. In general, how powerful does equipment need to be to handle different types of models? Based on this, I can determine what type of algorithms I can implement without crashing my computer. Also, how should I determine which methods to use? This is a good question to ask of a professional in an upcoming research interview, since they would have more experience about use cases for each method and what is easier to work with. Overall, my research has continued to progress and I am getting closer to starting to create something new!