# Object Detection with Transformers Project Proposal

Jazon Samillano

19 December 2020

## Overview

The use of Transformers to perform object detection on images is an exciting new endeavor in AI, and I am excited to dive into this topic head first. Only time will tell if Transformers are destined to kill off CNNs in object detection tasks - it's still way too early to tell. CNNs are still the gold standard in object detection when evaluated on the Common Objects in Context (COCO) dataset.

1. What is wrong with the current methods of using transformers on object detection?
    a. The average precision (AP) of Transformers in object detection is low relative to other state of the art solutions. On the COCO object detection test, the use of Transformers ranks only 17th on the list. This is using Facebook's **DETR** model. But let us please not forget that Transformers in object detection is still a little baby. When CNN was a baby, researchers did not yet know how to best architect them to achieve 90+ percent success on ImageNet, and it took several years to figure out how to get to that level. So Transformers being 17th on the state of the art list this early in the game is not something anyone should knock.
2. What is your idea in addressing this gap?
    a. Instead of trying to push higher on a generic dataset like COCO, I would like to fine-tune Facebook's DETR to become more specialized and more practical in a particular domain, such as self-driving cars.
3. Why do you think your idea will work or make sense?

   a. This idea came about when I tried fine-tuning Google’s depthwise separable convolutional model called **MobileNetV2**. I fine-tuned the model to start detecting skin cancer, and the results look great. I think I can fine-tune Facebook’s DETR to become really good at detecting objects on the road that are critical for the safe operation of self-driving cars.

4. What are the competing ideas on the same problem that you are attacking?
   a. Yolo v4, Yolo v5, and Faster R-CNN are really great at real-time object detections. I’m not sure how far Transformers will go in this domain, but I’d like to explore it further just to learn, in case Transformers later on become the gold standard for real-time object detections.
5. Why do you think your idea is better?
   a. The idea of fine-tuning DETR for a specific domain will bypass the very difficult task of becoming the world’s number one in generic object detections, such as in the COCO dataset. After all, being number one in COCO is not the reason why a self-driving car company would choose a particular model. Other serious considerations are in play, such as frames per second that the model can process on a real-time self-driving car camera feed. I don’t know if Transformers will one day slay its competitors in processing speed for camera feeds because it is way too early in the game. Yet, given the ferocity in which Transformers have troubled its competitors in the realm of NLP and image classification, I have a strong feeling Transformers may also soon rock the boat on older architectures in object detection. According to the arXiv.org paper “An Image is Worth 16x16 Words”, the incredible success the authors got with Transformers in image classification has not yet been replicated in object detection since object detection is a totally different beast. However, given the track record of Transformers, now is a great time to pour serious research effort into this if we want to be at the cutting-edge of computer vision deep learning.
6. How are we going to measure the performance of your proposed algorithm?
   a. I plan to compare my fine-tuned DETR to Google’s Faster R-CNN called **InceptionResNetV2** in detecting road objects critical to self-driving cars. I will then assess which path proves more promising. In other words, which model would a self-driving car company be more interested in. I hope it’ll be my fine-tuned DETR model.

7. Are there available public datasets that you can use? Are there current benchmarks that you can use to compare the performance of your algorithm? If none, are you going to make one?
    a. There are several datasets to choose from, such as the Waymo Open Dataset and the Lyft Level 5 Dataset. I am not aware of current benchmarks that I can use to assess the performance of my algorithm, but I will update this proposal in case I discover something new. The problem is that there might be a certain level of human finesse in assessing the performance. One example is something like, "Is it more important to detect a potted plant on the side of the road or an even smaller bag dropped by a pedestrian in the middle of the road. Well, it's nice for a model to detect the potted plant sitting on the side of the road, but I don't see how that helps the self-driving car operate more safely. A small bag in the middle of the road can mean the steering controller needs to slightly swerve because something in the bag might puncture the tires. A good model would be more practical if it detects the smaller bag on the road, even if it misses the potted plan on the side of the road. The opposite result of detecting the potted plant but missing the smaller bag would be considered an "impractical" result. I don't think there is an industry standard yet that can be used to make this type of "human" judgement. Another source of complication on this task is that self-driving car companies are notoriously secretive and would probably never publish the implementation or assessment details of how they moved their projects one step closer to being a Level 5 fully autonomous vehicle.
8. What are the metrics that you are going to use to measure the success/failure of your idea?
    a. If I can fine-tune DETR to be a more practical object detector for self-driving cars than Google's **InceptionResNetV2**, then I would label this a success. However, how will I define "more practical"? I am still baffled by this conundrum and may update this proposal as I get more mental clarity on that issue. It might end up being something subjective like, "If we asked Tesla or Waymo which is a more practical model, how will they respond?"

I look forward to working on this project and learning a ton in the process. I can formally swear that I did not get this idea of fine-tuning for self-driving cars from someone's Github or Medium article.  I arrived at this idea because I fine-tuned Google's **MobileNetV2** in TensorFlow a few months ago to detect skin cancer, and to top it off, I am wildly fascinated by people's ability to almost reach the point of Level 5 fully autonomous vehicles.  I am not yet committed on whether I should use TensorFlow or PyTorch.  It seems PyTorch is the natural choice given that DETR is Facebook's baby.  I'll dive into PyTorch a bit, but my comfort level is in TensorFlow, so I'd like to leave open the option of switching to TensorFlow if that proves more natural for me. Thank you for this opportunity.