Teaching an Object-Finding Model to Draw Maps and to Find the Least Information It Needs

Jiuen Feng Ruiying Zhu

Introduction

We took a model designed to find objects and taught it to draw a map of the road instead. The surprise: it needs remarkably little internal information, and bigger is not always better.

Main text

A car sees the world through several cameras. But separate side-on views are not enough: to drive safely, the car must fold them into a single map seen from above, showing which areas are open road and which are occupied.

Two families of models have grown up around this kind of perception. One family is designed to draw the map directly. It lays out a fixed three-dimensional grid of the space around the car and fills each cell by projecting it back into the camera images and reading off the matching pixels – a dense, fill-everything approach. The other family was invented for a different purpose: finding objects. Instead of filling a grid, it uses a small set of learned 'collectors', each one sent out to gather just the information it needs from across the camera views. This is a sparse approach – it deliberately keeps only a little.

We were curious whether that sparse, object-finding model could be taught to draw a map at all. The challenge is that a map is dense – it has a value for every location – while the collectors gather only sparse information. The model therefore has to unpack the little they bring back into a complete, dense map.

But drawing a map is not enough on its own. To be used in a real car, a model also has to be efficient – striking a balance between accuracy and cost. So we asked a practical question: how far can we shrink the collectors before the model can no longer unpack their information into a usable map? We ran this compression experiment on both families – the dense map-drawer and the sparse object-finder – measuring how map quality and computation changed as the internal sketch grew or shrank.

Our results show that bigger is not automatically better. For both models, enlarging the internal sketch helped only up to a point. Each had a useful range in which a fairly compact sketch matched the quality of much larger ones; beyond it, adding more brought computation but little benefit.

The reason is the same in both cases: past a certain size, extra capacity has no new information to capture. The model's structure and the camera images limit how much useful detail exists, so adding more collectors or a finer grid enlarges the sketch without enriching it. The collector-based model reached this ceiling from a surprisingly small sketch, since each collector covers a whole patch at once. The grid-based model behaved the same way, capped by the detail in its camera images.

Two concrete findings come out of this. First, a sparse, object-finding model can take on the harder, dense job of drawing a full map: the little its collectors gather really can be unpacked into a complete picture. Second, that compression has a clear balance point – a sketch size where accuracy and efficiency meet, below which the map degrades and above which the extra cost buys nothing. Finding that point is what lets such a system run lean enough for a real car without losing the detail it needs to map the road.