Introduction

We have created a large diverse set of cars from overhead images, which are useful for training a deep learner to binary classify, detect and count them. The dataset and all related material will be made publicly available. The set contains contextual matter to aid in identification of difficult targets. We demonstrate classification and detection on this dataset using a neural network we call RecPerception. This network combines residual learning with Inception-style layers and is used to count cars in one look. This is a new way to count objects rather than by localization or density estimation. It is fairly accurate, fast and easy to implement. Additionally, the counting method is not car or scene specific. It would be easy to train this method to count other kinds of objects and counting over new scenes requires no extra set up or assumptions about object locations.

Test patches the network correctly classified as containing a car in the central region. Occlusions and visibility issues are commonly handled, but we note that they still appear to account for much of the error. The left most image is not a mistake. It has a tree in the center while the shifted version above it has a car concealed slightly underneath the tree.

Detection

We tested detection on a held out set of 10 images comprising approximately 1 square km. Simple non-maximal suppression is used.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Count</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verification</td>
<td>260</td>
<td>253</td>
<td>9</td>
<td>7</td>
<td>96.65%</td>
<td>97.21%</td>
</tr>
<tr>
<td>Detection</td>
<td>260</td>
<td>253</td>
<td>10</td>
<td>10</td>
<td>92.59%</td>
<td>98.15%</td>
</tr>
</tbody>
</table>

Below can be seen several examples of detection on many scenes. Three of these are new scenes from post publication.

Network and Context

The percentage of correct patches v. the amount of context present. As more context is included, accuracy improves. It appears optimal to cut out a small amount of context.

Fast Counting

Examples of patches which were correctly counted by the network. From left to right the correct number is 9, 3, 6, 13 and 47. Note that cars which are not mostly inside a patch are not counted. The center of the car must be at least 8 pixels inside the visible region.

Fast counting takes in 224x224 patches in each stride which is 167 pixels. These are results on 20 test scenes like the ones above. The color borders show the extent of each patch/stride. The numbers in red are the count for each patch.

http://gdo-datasci.ucdlnl.org/cowc/