Automatically Gather Address Specific Dwelling Images Using Google Street View

Authors:
Salman Khan
Carl Salvaggio
OUTLINE

- Significance - Industry Challenges
- Method
- Results
Current Major Challenges in Industry

- Building Inventory upkeep
- Existing public record databases incomplete on old buildings
- Current policies provide little to no incentive in sharing of privately owned data
- Expensive and lengthy process in developing accurate building inventory
Cannot rely on images straight from the Google Street View API (Google SV API)
AutoCrop

Raw Google SV API Images

Final Cropped Images
Employed 3 types of neural networks:

- **Scene Classification**
  - Pytorch-ResNet-50 pre-trained model - Places-365 data set

- **Object Detection**
  - Keras-RetinaNet pre-trained model - Google’s OpenImagesV4 data set

- **Semantic Segmentation**
  - Keras-PSPNet-101 pre-trained model – Cityscape data set
AutoCrop – Scene Classification

- Accepted images from scene classification
AutoCrop – Object Detection

- Object Detection

1. Get object detection results on an image and its flipped version.
2. Flip back boundary boxes of flipped image onto original orientation.
3. Merge matching boxes.
4. Run loop over each combination of boundary boxes.
5. Calculate IoBA for every combination.

Void selection if:
- ROI < Area of non-dwelling type or IoU >= 0.6 with non-dwelling box.

Select boundary box for crop.
- Keep boundary box with higher score. Delete the other.
- Keep boundary boxes.

Combinations where IoBA >= 0.9?
AutoCrop – Object Detection

Fig. 15: Object Detection Results of Networks Trained on 443 Labels Only (showing top 5 matches for each label to reduce clutter)

(a) RetinaNet with ResNet50 backbone (b) RetinaNet with ResNet101 backbone (c) RetinaNet with ResNet152 backbone

Fig. 16: Object Detection Results of Networks Trained on All Labels (showing top 5 matches for each label to reduce clutter)

- Individual Network Results
AutoCrop – Object Detection

- Merged Result
AutoCrop – Object Detection

▪ After our NMS Strategy
AutoCrop – Object Detection

- IoBA NMS Algorithm

**Algorithm 1 Bounding Box Reduction via Intersection over smaller Box Area (IoBA)**

Read in Bounding Boxes (BB), Labels (L) and Scores (S)

for UniqueLabel do

  Combination (C) of every index

  if Del not in C then

    for Indices in C do

      Calculate area of boxA and boxB

      IoBA = Intersection / min(boxA, boxB)

      if IoBA >= 0.90 then

        Take BB with higher S

        Register deletion (Del) index

      end if

    end for

  end if

end for
AutoCrop – Object Detection

- Per label, make combination of indices
- Per combination, calculate: $\text{IoBA} = \frac{\text{Area of Intersection}}{\text{min(box-Area)}}$
AutoCrop – Object Detection

Selected Region of Interest
- **Semantic Segmentation**

1. Get Semantic Segmentation Results on ROI
2. More than 50% Tree Coverage?
   - Yes: Reject Image
   - No: Crop image to size of selected boundary box
3. All Images predicted?
   - Yes: End
   - No: 1
AutoCrop – Semantic Segmentation

- Images removed from final selection
### Final Empirical Results of AutoCrop

- **TP**: `Final Cropped Images' - `Not ROI'
- **FP**: `Not ROI'
- **FN**: Misclassified images by Scene Classification Network + Images rejected by Semantic Segmentation but visually appear to have less than 50% tree occlusion

<table>
<thead>
<tr>
<th>City</th>
<th>Total Images</th>
<th>Accepted Scenes</th>
<th>Post NMS</th>
<th>Final Cropped Images</th>
<th>Not ROI</th>
<th>Err/ Prc/ Rcl</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF Bay Area</td>
<td>321</td>
<td>258</td>
<td>136</td>
<td>116</td>
<td>1</td>
<td>0.86%/ 99.14%/ 83.33%</td>
</tr>
<tr>
<td>Seattle</td>
<td>1106</td>
<td>749</td>
<td>453</td>
<td>405</td>
<td>1</td>
<td>0.25%/ 99.75%/ 84.34%</td>
</tr>
<tr>
<td>Portland</td>
<td>1620</td>
<td>993</td>
<td>587</td>
<td>579</td>
<td>3</td>
<td>0.52%/ 99.48%/ 74.32%</td>
</tr>
<tr>
<td>Santa Monica</td>
<td>1490</td>
<td>1191</td>
<td>322</td>
<td>257</td>
<td>4</td>
<td>1.57%/ 98.44%/ 71.07%</td>
</tr>
<tr>
<td>Vancouver</td>
<td>726</td>
<td>569</td>
<td>349</td>
<td>285</td>
<td>1</td>
<td>0.35%/ 99.65%/ 81.14%</td>
</tr>
</tbody>
</table>

| Averaged Results: | Avg Err: | 0.428% | Avg Prc: | 99.29% | Avg Rcl: | 78.84% |

- **Precision** = **TruePositives** / (**TruePositives** + **FalsePositives**)
- **Recall** = **TruePositives** / (**TruePositives** + **FalseNegatives**)

**AutoCrop – Empirical Analysis**
### AutoCrop – Empirical Analysis

<table>
<thead>
<tr>
<th>City</th>
<th>Total Images</th>
<th>Accepted Scenes</th>
<th>Post NMS</th>
<th>Final Cropped Images</th>
<th>Not ROI</th>
<th>Err/ Prc/ Rcl</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF Bay Area</td>
<td>321</td>
<td>258</td>
<td>256</td>
<td>217</td>
<td>14</td>
<td>6.45%/93.55%/89.82%</td>
</tr>
<tr>
<td>Seattle</td>
<td>1106</td>
<td>749</td>
<td>673</td>
<td>475</td>
<td>23</td>
<td>4.84%/95.15%/84.96%</td>
</tr>
<tr>
<td>Portland</td>
<td>1620</td>
<td>993</td>
<td>954</td>
<td>704</td>
<td>59</td>
<td>8.34%/91.61%/73.13%</td>
</tr>
<tr>
<td>Santa Monica</td>
<td>1490</td>
<td>1191</td>
<td>1124</td>
<td>860</td>
<td>191</td>
<td>22.20%/77.79%/86.43%</td>
</tr>
<tr>
<td>Vancouver</td>
<td>726</td>
<td>569</td>
<td>558</td>
<td>348</td>
<td>14</td>
<td>4.02%/95.97%/82.2%</td>
</tr>
<tr>
<td><strong>Averaged Results:</strong></td>
<td><strong>Avg Err:</strong> 9.17%</td>
<td><strong>Avg Prc:</strong> 90.81%</td>
<td></td>
<td></td>
<td><strong>Avg Rcl:</strong> 83.31%</td>
<td></td>
</tr>
</tbody>
</table>
## Empirical analysis on thresholds

<table>
<thead>
<tr>
<th>City</th>
<th>$Occ_{Thr_{0.6}}$ &amp; $IoBA_{0.85}$</th>
<th>$Occ_{Thr_{0.6}}$ &amp; $IoBA_{0.95}$</th>
<th>$Occ_{Thr_{0.55}}$ &amp; $IoBA_{0.9}$</th>
<th>$Occ_{Thr_{0.65}}$ &amp; $IoBA_{0.9}$</th>
<th>$Occ_{Thr_{0.6}}$ &amp; $IoBA_{0.9}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF Bay Area</td>
<td>1/127 = 0.79%</td>
<td>1/127 = 0.79%</td>
<td>1/122 = 0.79%</td>
<td>1/122 = 0.79%</td>
<td>1/116 = 0.86%</td>
</tr>
<tr>
<td>Seattle</td>
<td>16/411 = 3.89%</td>
<td>5/403 = 1.20%</td>
<td>6/411 = 1.46%</td>
<td>6/411 = 1.46%</td>
<td>1/405 = 0.25%</td>
</tr>
<tr>
<td>Portland</td>
<td>4/548 = 0.73%</td>
<td>4/540 = 0.74%</td>
<td>4/545 = 0.74%</td>
<td>4/545 = 0.74%</td>
<td>3/579 = 0.52%</td>
</tr>
<tr>
<td>Santa Monica</td>
<td>9/279 = 3.22%</td>
<td>12/278 = 4.31%</td>
<td>9/279 = 3.22%</td>
<td>9/279 = 3.22%</td>
<td>4/257 = 1.57%</td>
</tr>
<tr>
<td>Vancouver</td>
<td>3/303 = 0.98%</td>
<td>3/303 = 0.99%</td>
<td>3/308 = 0.97%</td>
<td>3/308 = 0.97%</td>
<td>1/285 = 0.35%</td>
</tr>
<tr>
<td>Avg Error</td>
<td>1.92%</td>
<td>1.61%</td>
<td>1.44%</td>
<td>1.44%</td>
<td>0.428%</td>
</tr>
</tbody>
</table>