Predicting ground-level scene Layout from Aerial imagery

Muhammad Hasan Maqbool
Objective

• Given the overhead image → predict its ground level semantic segmentation

Overhead/Aerial Image

Predicted ground level labeling

Ground Level Image
Overall Architecture (Training)

- Overhead/Aerial Image
  - VGG
  - Labeling
  - Transform
  - Ground Level Labeling (Predicted)
- Ground Level Image
  - SegNet
  - Cross Entropy Loss
  - Ground Level Labeling
Ground Truth of Ground-level Segmentation

1. Generate the segmentation of ground level image using (pre-trained model)
   1. SegNet → This gives “Ground Truth” for training
2. To avoid manual pixel annotation of their new dataset (overhead imagery)
Algorithm Overview
Aerial Image to Ground Level (Predicted) Segmentation

128 Conv 1 x 1
64 Conv 1 x 1
1 Conv 1 x 1
Algorithm Overview

Predicted and Ground Truth Layouts Distance Minimization

• Minimize the pixel distance between the ground layout and the aerial-to-ground layout prediction

Ground Level Layout (ground truth)  Predicted Ground Level Layout

Cross Entropy Loss
Network A

VGG16

Interpolate to size 17 x 17 x channels

Hypercolumn 17x17x960

Three conv layers employ 1x1 convolution
Reduce channel dimension from 960 to 4
Network S

Three conv layers employ 1x1 convolution
Reduce channel dimension from 512 to 1

Encode the global information of the aerial image
Determine the Size of the Transformation Matrix

- Aerial image layout $l_a$ is multiplied by transformation matrix $M$ to get the predicted ground-level layout $l_g$

Based on the input size and the desired output size, we can determine the size of $M$ ----> In this case, we have $M = 320 \times 289$
Construct Input to Network F

S network

17x17x1
Reshape
1x1x289

Width (W) = 289
Height: H = 320
Reshape
Width (W) = 289
Height: H = 320
Concatenate coordinates
(i, j, y, x)
1x1x4

How to set the coordinates for each pixel will be explained later

How to set the coordinates for each pixel will be explained later

Concatenate coordinates

Width (W) = 289
Height: H = 320
Channel: C = 289 + 4 = 293

How to set the coordinates for each pixel will be explained later

Concatenate coordinates

Width (W) = 289
Height: H = 320
Channel: C = 289 + 4 = 293

How to set the coordinates for each pixel will be explained later

Concatenate coordinates

Width (W) = 289
Height: H = 320
Channel: C = 289 + 4 = 293

How to set the coordinates for each pixel will be explained later

Concatenate coordinates

Width (W) = 289
Height: H = 320
Channel: C = 289 + 4 = 293

How to set the coordinates for each pixel will be explained later

Concatenate coordinates

Width (W) = 289
Height: H = 320
Channel: C = 289 + 4 = 293

How to set the coordinates for each pixel will be explained later

Concatenate coordinates

Width (W) = 289
Height: H = 320
Channel: C = 289 + 4 = 293

How to set the coordinates for each pixel will be explained later

Concatenate coordinates

Width (W) = 289
Height: H = 320
Channel: C = 289 + 4 = 293

How to set the coordinates for each pixel will be explained later

Concatenate coordinates

Width (W) = 289
Height: H = 320
Channel: C = 289 + 4 = 293

How to set the coordinates for each pixel will be explained later

Concatenate coordinates

Width (W) = 289
Height: H = 320
Channel: C = 289 + 4 = 293

How to set the coordinates for each pixel will be explained later

Concatenate coordinates
Network F

Width (W) = 289

Channel: $C = 289 \times 4 = 293$

Height: $H = 32C$

Transformation matrix

Network F

320 x 289 x 128

320 x 289 x 64

320 x 289 x 1
How to determine the coordinates?

(i, j, y, x)

Width (W) = 289

Height: $H = 32c$

Channel: $C = 289 + 4 = 293$
How to determine the coordinates?

Recall – the following procedure is used to get the predicted ground-level layout:

Transformation matrix $M$ times the vector $\text{Vec}(17 \times 17 \times 1)$ equals $8 \times 40 \times 1$. The result is reshaped.
How to determine the coordinates?

How to determine the coordinates?

i: row index
j: column index

(i, j) = (0, 0)
(i, j) = (0, 16)
(i, j) = (1, 0)
(i, j) = (1, 16)
(i, j) = (16, 16)

Vec (i, j) = (16, 16)

289-D vector
How to determine the coordinates?

\[ \text{Vec} \left( I_a \right) \times \left( \begin{array}{c} 17 \times 17 \times 1 \end{array} \right) = \text{320-D vector} \]

\[ X_{8 \times 40} \]

\[ y: \text{row index} \]
\[ x: \text{column index} \]
How to determine the coordinates?

320-D vector

(y, x) = (0, 0)

(y, x) = (0, 39)

(y, x) = (1, 0)

(y, x) = (1, 39)

(y, x) = (7, 39)

Reshape

x: column index

y: row index

Ig

8x40
How to determine the coordinates?

(i, j) = (0, 0)
(i, j) = (0, 16)
(i, j) = (1, 0)
(i, j) = (1, 16)
(i, j) = (16, 16)

(y, x) = (0, 0)
(y, x) = (0, 39)
(y, x) = (1, 0)
(y, x) = (1, 39)
(y, x) = (16, 39)

0th row
39th row

(I, j, y, x) = (0, 0, 0, 0)
(I, j, y, x) = (0, 1, 0, 0)
(I, j, y, x) = (0, 16, 0, 0)
(I, j, y, x) = (1, 0, 0, 0)
(I, j, y, x) = (1, 39, 0, 0)

One row determines a single pixel value at location (y, x)

320-D (rotate to a row vector due to space limit)
How to determine the coordinates?

(i, j) = (0, 0)
(i, j) = (0, 16)
(i, j) = (1, 0)
(i, j) = (1, 16)
(i, j) = (16, 16)

(y, x) = (0, 0)
(y, x) = (0, 39)
(y, x) = (1, 0)
(y, x) = (1, 39)
(y, x) = (16, 39)

One row determines a single pixel value at location (y, x)
Interpret the transformation matrix

One row determines a single pixel value at location (y, x)

This is

Weight the 17x17 pixels in Ia

Determine one pixel in Ig

Each row of the transformation matrix e.g.

M 320x289

(i, j) = (0, 0)
(i, j) = (0, 16)
(i, j) = (1, 0)
(i, j) = (1, 16)

(16, 16)
Transformation Matrix Visualization

• The transformation matrix encodes the relationship between the aerial-level pixels and the ground-level pixels.
Dataset
Cross-View Dataset

• CVUSA contains
  • approximately 1.5 million geo-tagged pairs of ground and aerial images from across the United States.
  • Google Street View panoramas of CVUSA as our ground level images
  • Corresponding aerial images at zoom level 19
  • 35,532 image pairs for training and 8,884 image pairs for testing.
Cross-View Dataset: An Example

In the aerial images, north is the up direction.
In the ground images, north is the central column.
Cross-View Dataset
Cross-View Dataset
Evaluation and Applications
Weakly Supervised Learning
Weakly Supervised Learning for Aerial Image Classification and Segmentation
As Pre-trained Model
Network Pre-training for Aerial Image Segmentation

[Images of aerial images and their segmentations]
Network Pre-training for Aerial Image Segmentation

Per-class Precision on the ISPRS Segmentation Task

<table>
<thead>
<tr>
<th>Class</th>
<th>Init.</th>
<th>Number of training samples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>0.67</td>
<td>0.74</td>
</tr>
<tr>
<td>Imp.</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td></td>
<td>0.60</td>
<td>0.55</td>
</tr>
<tr>
<td>Ours</td>
<td>0.72</td>
<td>0.76</td>
</tr>
<tr>
<td>Bldg</td>
<td>0.56</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td>0.78</td>
<td>0.72</td>
</tr>
<tr>
<td>Ours</td>
<td>0.37</td>
<td>0.43</td>
</tr>
<tr>
<td>Low.</td>
<td>0.29</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>Ours</td>
<td>0.68</td>
<td>0.54</td>
</tr>
<tr>
<td>Tree</td>
<td>0.42</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>0.36</td>
<td>0.44</td>
</tr>
<tr>
<td>Ours</td>
<td>0.13</td>
<td>0.46</td>
</tr>
<tr>
<td>Car</td>
<td>0.05</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.11</td>
</tr>
</tbody>
</table>
Network Pre-training for Aerial Image Segmentation
Orientation Estimation
Orientation Estimation

• Given a street view image $I_g$, estimate its orientation
  1. Get the corresponding aerial image (using GPS tag), $I_a$
  2. Infer the semantic labeling of the query image, $I_g \rightarrow L_g$
  3. Predict the ground image labels from the aerial image using the proposed network, $I_a \rightarrow L'_g$
  4. Compute the cross entropy between $L_g$ and $L'_g$ in a sliding window fashion across all possible orientations
  5. The smallest entropy is the optimal estimated orientation
Orientation Estimation

• Their CVUSA dataset contains aligned image pairs

• Therefore, the learned network predicts the ground (panorama image) layout in a fixed orientation – served as reference
Orientation Estimation

Query street view image

Get segmentation (e.g. SegNet)

Get aerial image

Get predicted ground-level layout using the proposed network

Predicted ground-level layout of the panorama image

W

H

1 column = 360 degree / W

Compute cross-entropy in sliding window fusion
Orientation Estimation
Geocalibration
Fine-Grained Geocalibration
Fine-Grained Geocalibration
Ground Level Image Synthesization
Ground-level Image Synthesization
Ground-level Image Synthesization
Conclusion

• A novel strategy to relate aerial-level imagery and the ground-level imagery

• A novel strategy to exploit the automatically labeled ground images as a form of weak supervision for aerial imagery understanding

• Show the potential of this method in geocalibration and ground appearance synthetization
Questions