JOEL JURIK

Report Meeting – 6/12

Project 1

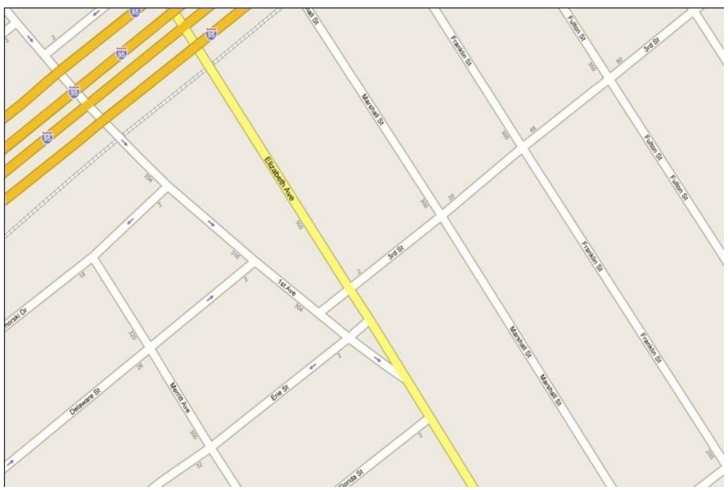
Using Google Maps to Improve Detection of Objects from Satellite Images

Project Outline

- Problem: Detection is difficult from satellite images due to the object's small size
- Solution: Improve detection by identifying the roads in Google maps that correspond to the satellite image, then using that information in order to limit where to look for objects

Project Outline

1. Use position (latitude/longitude) of satellite image to query Google maps



2. Extract the roads.

Project Outline

3. Match up the roads with satellite image

4. Use this information to search for objects, such as cars or buildings.

Initial Plan

- By 6/19: Install and learn basics of OpenCV, look more into the Google maps API, annotate test data, begin discussing with Mikel about object detection.
- By 7/3: Use current methods for object detection to learn how everything works, begin creating algorithm to line-up roads with satellite data.
- By 7/10: Complete algorithm for lining up roads with satellite data, begin integrating algorithm with existing methods for object detection.

Initial Plan

- By 7/24: Complete integration of algorithm with methods for object detection, begin testing and evaluation, begin thinking about research paper.
- By 8/7: Done with research paper that gives the results.

Progress

- Looked into the Google API
- Read “Learning Spatial Context: Using Stuff to Find Things” by Heitz and Koller
- Installed and learning OpenCV

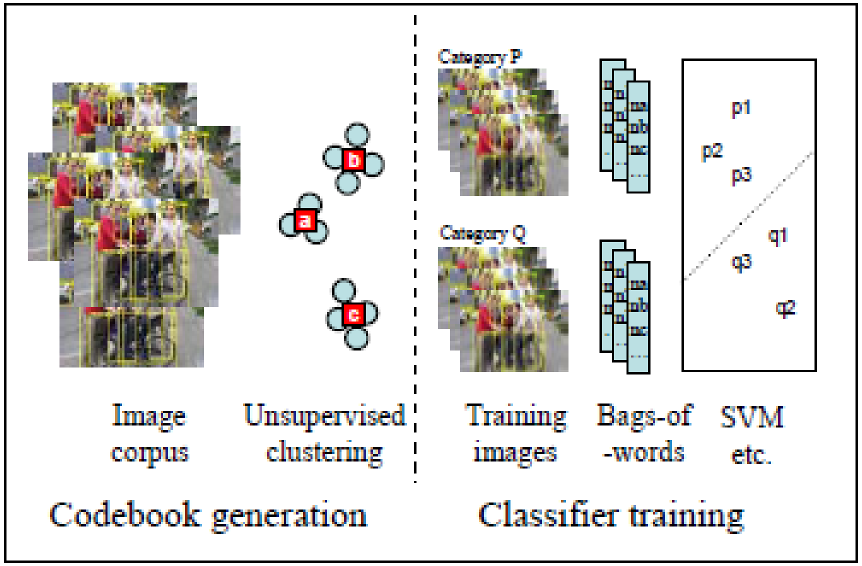
Project 2

Visual Bits

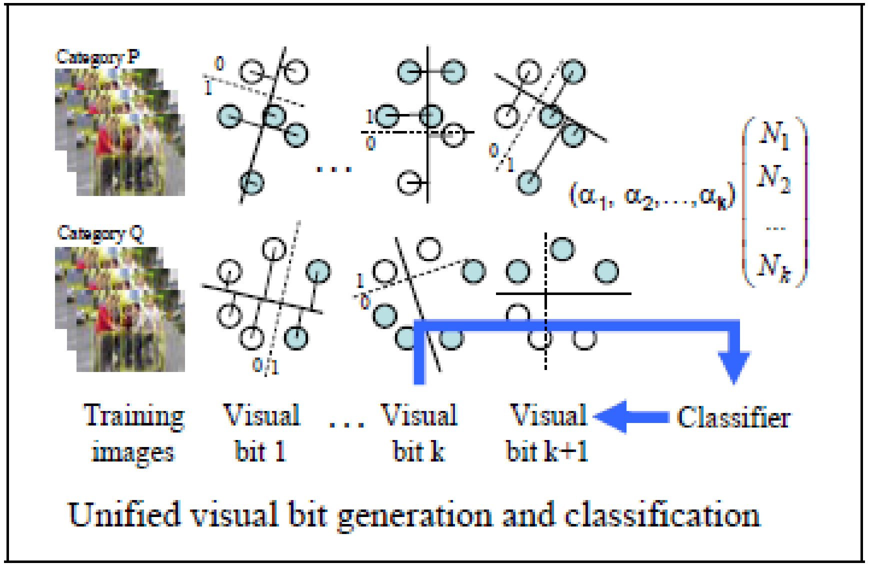
Project Outline

- Problem: Existing techniques separate codebook generation and classifier training. Also, the relationship between visual words forces the relationship between two features to be all-or-none. This involves complicated math.
- Solution: Find intuitive way to solve the problem, but without the complicated math.

Project Outline



Standard Approach



Visual Bits Approach

(from Yang, Jin, Sukthankar, and Jurie)

Project Outline

- Visual bit generated by thresholding the image feature along a category-specific direction
- Image represented by a set of vectors, one for each object category
- Classifier is trained for each category by processing the vector and indicate the degree in which in matches the category.

Project Outline

Minimize objective function:

$$\mathcal{L}' = \sum_{i=1}^{N} \sum_{y \in \mathbf{y}_i} l'(X_i, y) = \sum_{i=1}^{N} \sum_{y \in \mathbf{y}_i} \frac{n_i}{\sum_{j=1}^{n_i} l'(\mathbf{x}_{i,j}, y)}$$

Separate combination weight and then minimize the upper bound of this equation:

$$\mathcal{L}' \leq \frac{e^{-3\alpha} + e^{3\alpha} + 1}{3}\mathcal{L} - \frac{1 - e^{-3\alpha}}{3} \sum_{y=1}^{m} \sum_{i=1}^{N} \sum_{j=1}^{n_i} g(\mathbf{x}_{i,j}, y) T_{i,j}^{y},$$

where

$$T_{i,j}^{y} = \sum_{y' \in \mathbf{y}_i} l(X_i, y') q_{i,j}(y') [\delta(y, y') - e(\mathbf{x}_{i,j}, y))].$$

Find combination weight by:

$$\alpha = \frac{1}{2} \log \left(\frac{\sum_{i=1}^{N} \sum_{j=1}^{n_i} A_{i,j}(0) B_{i,j}(1)}{\sum_{i=1}^{N} \sum_{j=1}^{n_i} A_{i,j}(1) B_{i,j}(0)} \right),$$

where

$$A_{i,j}(z) = \sum_{y=1}^{m} e(\mathbf{x}_{i,j}, y) \delta(g(\mathbf{x}_{i,j}, y), z)$$

$$B_{i,j}(z) = \sum_{y \in \mathbf{y}_i} l(X_i, y) q_{i,j}(y) \delta(g(\mathbf{x}_{i,j}, y), z)$$

Find a new way that isn't so computationally complex!

Initial Plan

- By 6/19: Have code from Dr. Sukthanker and begin looking at it and thinking of ideas.
- By 6/26: Understand code.
- By 7/17: Come up with a few ideas to try to reduce the math, but with same results.
- By 7/26: Come up with working idea and begin implementation.
- By 8/3: Begin paper.
- By 8/10: Finish paper with results.

Progress

- Learning about SVMs, RBF, SIFT, Optimization.
- Emailed Dr. Sukthanker for code.