Stealing Objects
With Computer Vision

Learning Based Methods in Vision
Analysis Project #4: Mar 4, 2009
Presented by: Brian C. Becker
Carnegie Mellon University
Motivation

• Goal: Detect objects in the photo you just took
Motivation

- Scanning window
Motivation
Motivation
Motivation
Motivation
Motivation
Motivation
Motivation

• What else can we try for object recognition?
Object Detection

- Go to internet and behold! exact picture labeled with:
  - Sky
  - Tower
  - Sky Tower
  - Building
  - Tree
  - Road
  - Traffic Light
  - Windows
  - Window
  - Arch
  - Car
  - Lampost
  - Street Sign
  - Person
  - Door
  - Car
  - Road
Object Detection

• Ideally, object detection is giant lookup
  – Labeled plenoptic function
  – Label everything in the world from all viewpoints

• Labelme: Online annotation tool
Tool went online July 1st, 2005
290,000 object annotations

Labelme.csail.mit.edu

B. Russell, A. Torralba, K. Murphy, W.T. Freeman. IJCV 2008
Labelme Polygon Quality
Labelme Polygon Diversity

Paper cup

Rock

Statue

Chair
Labelme Testing

Most common labels:
- test
- adksdsa
- woiieiie
- ...

[Images of various objects with labeled areas]
Labelme Hooligans
Do not try this at home
Labelme Database

• 30 GB dataset of
  – 176,000 photos total
  – 52,000 photos with annotations
Labelme Matlab Toolbox

LMquery (database, 'object.name', 'car,building,road,tree')

• Query objects
• Extract polygons
• Annotation stats
• Label merging
• Wordnet reasoning
• Manipulate images
• Scene descriptors
Wordnet Object & Parts
Object Detection

• Unfortunately, Labelme is not God
• Next best thing
  – Find similar scenes containing similar objects
  – Steal information from them (i.e. label transfer)
Papers

• SIFT Flow Paper

• Context Paper
SIFT Flow

• SIFT Flow Goal: Align objects in similar scenes
  – Problem: Current alignment algorithms aren’t robust
  – Solution: SIFT is magic and works, find the flow of image patches to a similar image

• If your dataset isn’t infinite, find a close match and rearrange (wiggle) it so it is aligned

• SIFT Flow “allows matching of objects located at different parts of the scene”
SIFT Flow

Input image

• Labels
• Motion
• Depth
• ...

The space of world images

Nearest neighbors

• Labels
• Motion
• Depth
• ...

Matching SIFT Features

- Decompose image into scene descriptors
- SIFT features (D. Lowe, 1999)
  - 128 dimensional vector \((u_1, \ldots, u_{128})\) at each pixel
Matching SIFT Features

- Use “bag-of-words” to cluster SIFT features into 500 visual words
  - Good ole K-means
- Reduce image to texton map of SIFT features
- Fast/coarse matching on SIFT texton map
- Top 20 fast matches re-ranked with SIFT Flow
SIFT Flow

• Optical flow without spatial limitations
• Assumptions:
  – SIFT descriptors at each pixel are constant with respect to the pixel displacement field
  – One pixel may move as much as the size of the image
  – Grouping of pixels (move clusters of pixels)
SIFT Flow

• Formulate as an optimization problem

\[
E(w) = \sum_p \| s_1(p) - s_2(p + w) \|_1 + \frac{1}{\sigma^2} \sum_p \left( u^2(p) + v^2(p) \right) + \\
\sum_{(p,q) \in \varepsilon} \min \left( \alpha |u(p) - u(q)|, d \right) + \min \left( \alpha |v(p) - v(q)|, d \right),
\]

– \( w(p) = (u(p), v(p)) \) is the displacement vector at pixel location \( p = (x, y) \)

– \( S_i(p) \) is the SIFT descriptor extracted at location \( p \) in image \( I \)

– \( E \) is the spatial neighborhood of a pixel
SIFT Flow

- Formulate as an optimization problem

\[ E(w) = \sum_{p} \|s_1(p) - s_2(p + w)\|_1 + \frac{1}{\sigma^2} \sum_{p} \left( u^2(p) + v^2(p) \right) + \sum_{(p,q) \in \varepsilon} \min \left( \alpha|u(p) - u(q)|, d \right) + \min \left( \alpha|v(p) - v(q)|, d \right), \]

- \( u \) and \( v \) are decoupled to reduce complexity from \( O(L^3) \) to \( O(L^2) \). \( L \) is the size of the search window.
SIFT Flow Example

• SIFT Flow “allows matching of objects located at different parts of the scene”

• Hypothesis: Pixels from an object in one image will “flow” to the same class of objects in a second image

• Let’s test that with a simple example
**SIFT Flow Pepper Example**

- Two images of a pepper
  - One pepper is shifted 20 pixels right, 10 pixels up

First image warped to second
SIFT Flow Pepper Example

- Two images of a pepper
  - One pepper is shifted 100 pixels right, 50 pixels up

- Test turning off continuity

- Needs lot of tweaking
SIFT Flow Hard Example
SIFT Flow Hard Example

- Felzenszwalb parts-based HOG detector says -1.0289 score
SIFT Flow Hard Example
SIFT Flow Hard Example

• Best match, most similar labeled photo
SIFT Flow Hard Example
SIFT Flow Hard Example
SIFT Flow Hard Example

Match

Query

Turn off continuity
SIFT Flow Hard Example

Y flow amount

Match

Query
SIFT Flow Hard Example

New Query

Same Match
SIFT Flow Hard Example
SIFT Flow Hard Example

Y flow amount

Query

Match
SIFT Flow Paper Examples
SIFT Flow Paper Examples
Estimating Motion

• What else can we do with SIFT Flow?
Motion Ambiguity

• Multiple plausible motions
Synthesizing Motion

Input Image  Composite Video  Retrieved Motion
Papers

• SIFT Flow isn’t quite there yet

• If you can’t match objects in images
  – Find similar, but non-spatially aligned scenes
  – Use labeled information as a prior

• Context Paper
Object Detection

- Use a “context-enhanced” sliding window
- Retrieve K similar scenes and extract priors
  - Frequency and spatial information
  - Weaker form of label transfer based on “clues”
Context Approach

• Goal: Recognize objects embedded in a scene
Retrieval set + LabelMe labels

- Steal object
  - Frequency
  - Location
  - Size
  - Etc
Goals

• Given $db$: A database of labeled images
• Given $img$: A new image

• Find images similar to $img$ in $db$
  – Similar scenes (mountain, office, etc)
  – Similar objects (coffee cup, car, etc)
  – Similar layout (lake on left, building to right)

• Basically, scene alignment
Matching Gist Features

- Decompose image into scene descriptors
- Gist features (A. Oliva, et. al. 2001)
Matching Gist Features

- Apply oriented Gabor filters over different scales
- Average filter energy in each bin

- Used for scene recognition
- Similar to SIFT (Lowe 1999)

8 orientations
4 scales
\[ \times 16 \] bins
512 dimensions
Evaluation Dataset

• Used a subset of the Labelme dataset

• Training:
  – 15,691 images
  – 105,034 labels

• Testing:
  – Cities/offices outside of training set
  – 560 images
Predicting Object Presence

- Can descriptor predict the presence of

  Does this image contain:
  - Person?
  - Computer monitor?
  - Building?
  - Beer?
  - Car?
  - Etc...

- Or use indirect method of matching images

  Do these images contain:
  - Person?
  - Computer monitor?
  - Building?
  - Beer?
  - Car?
  - Etc...
SVM Object vs. kNN

- Per object SVM
  - SVM trained on object bounding box gist features
  - SVM applied to bounding boxes in image
  - Maximal score used

- Retrieval set:
  - Histogram object labels
  - Use normalized histogram value to classify image
Method/k Comparison

AUC vs. # of images matched (30 classes)

- 512D L1 GIST
- SIFT Flow a=0
- SIFT Flow a=2
- 16x16 L2 LAB
- 200D SIFT texton 2D Spatial Pyramid

Area Under Curve for ROC

Size of Retrieval Set (30 classes)
SVM (image) vs. kNN

SVM vs. NN on gist features (100 most common train objects)

ROC AUC NN on gist features
ROC AUC SVM trained on gist features

Rock, river, painting, curtain, carpet, sand, cloud, pillow, vase, TV, faucet, brushes, eye, snow, ceiling light

One vs. all SVM on image descriptors
kNN on GIST with L1 distance metric
Method/k Comparison

AUC vs. # of images matched (100 classes)

- 512D L1 GIST
- SIFT Flow a=0
- SIFT Flow a=2
- 16x16 L2 LAB
- 200D SIFT texton 2D Spatial Pyramid

Area Under Curve for ROC

Size of Retrieval Set (100 classes)
Using Retrieval Set

• Object detection uses variable-sized sliding windows and an SVM appearance model
  – Very slow, ~4,000 bboxes to calculate gist for

• Find contextual clues in retrieval set
  – If all the matched images were of streets, unlikely to find a keyboard

• Build a probabilistic model including information transferred from matched images
Using Retrieval Set

• Probabilistic Formulation

\[
p(o, x, g | \theta, \phi, \eta) = \prod_{i=1}^{N} \prod_{j=1}^{M_i} \sum_{h_{i,j}=0}^{1} p(o_{i,j} | h_{i,j}, \theta) p(x_{i,j} | o_{i,j}, h_{i,j}, \phi) p(g_{i,j} | o_{i,j}, h_{i,j}, \eta)
\]

– N images, M object proposals per image, L classes
– \( h_{i,j} = 1 \) indicates object class \( o_{i,j} \) is present at location \( x_{i,j} \)
Using Retrieval Set

• Probabilistic Formulation

\[ p(o, x, g | \theta, \phi, \eta) = \prod_{i=1}^{N} \prod_{j=1}^{M_i} \sum_{h_{i, j}=0}^{1} p(o_{i, j} | h_{i, j}, \theta) p(x_{i, j} | o_{i, j}, h_{i, j}, \phi) p(g_{i, j} | o_{i, j}, h_{i, j}, \eta) \]

– Spatial locations encoded by centroid & size of bounding box of object (normalized to [0,1])

– Probability parameters and are learned from the retrieval set on \( \theta_m, \phi_{m,l} \)

– Probability parameter is learned offline by training an SVM for each object class on training set
Using Retrieval Set

• Advantages
  – Can increase accuracy if retrieval set is good
  – Can save CPU time by constraining search
    • Look only for objects likely to be in the image
    • Look only for objects in likely locations

• Disadvantages
  – Can decrease accuracy if retrieval set is bad
  – Non-exhaustive search can miss objects
    • Maybe there is a bike indoors
Context Approach

- Goal: Recognize objects embedded in a scene

Input image

Nearest neighbors from 15,691 images

Cluster images using object labels

Output image with object labels transferred
Clustering Retrieval Set

Cluster images based on labels:

- Object identity
- Location within image
Clustering Retrieval Set

• “Used a simple model to cluster object labels belonging to the retrieved images”

• Incorporate latent clusters with mixing weights
• Cluster object labels and spatial locations
• Dirichlet process prior with stick-breaking
• Rao-Blackwellized Gibbs sampler
• Manually tuned hyperparameters
• Perform hard Expectation Maximization (EM)
Clustering Retrieval Set

$\alpha \rightarrow \pi \rightarrow s_i \rightarrow M_i \rightarrow o_{ij} \rightarrow x_{i,j} \rightarrow \phi_{k,l}$

$S_i$ - cluster assignment

$O_{ij}$ - object labels

$x_{ij}$ - bounding box parameters

$s_i | \pi \sim \pi$

$\pi | \alpha \sim Stick(\alpha)$

$o_{i,j} | s_i = k, \theta \sim \theta_k$

$\theta_k | \beta \sim Dirichlet(\beta)$

$x_{i,j} | s_i = k, o_{i,j} = l, \phi \sim \mathcal{N}(\phi_{k,l})$

$\phi_{k,l} | \gamma \sim \mathcal{NIW}(\gamma)$
Clustering Retrieval Set

Use Gibbs sampler to draw scene assignments:

\[ s_i \sim p(s_i \mid s_{\setminus i}, o, x, \alpha, \beta, \gamma) \]

Chinese restaurant process analogy:
tables - scene parameters; customers - images
Results: ROC Curves

Boxplot of Area Under Curve of ROC

- SVM
- No Clustering
- Clustering

Clustering hurts performance!
Context Approach

• Goal: Recognize objects embedded in a scene

Input image

Nearest neighbors from 15,691 images

Cluster images using object labels

Output image with object labels transferred
Outputs
Results: ROC Curves

Blue: SVM       Red: No Clustering     Green: With Clustering
Results: ROC Curves

Blue: SVM       Red: No Clustering     Green: With Clustering
Pascal 2007 Results
Pascal 2007 Results
Pascal 2007 Results
Pascal 2007 Results

AUCs vs. # of images matched (20 classes)
Pascal 2007 Results

AUC vs. # of images matched (20 classes)
Summary

• Stealing is good and helps your accuracy

• SIFT Flow tries to solve the finite data problem
  – Morph images so they do match perfectly
  – Decent idea, but needs more work

• Context transfers info from similar images
  – Small but noticeable improvements
  – How much data do you need?
Conclusion

- Context is yet another knob to tweak