Nonparametric Scene Parsing: Label Transfer via Dense Scene Alignment
Basic Idea

From Liu, Yuen, Torralba

(a) Input Image

(b) Top Matches

(c) Annotations of top matches

(d) Output

(e) Ground Truth

From Liu, Yuen, Torralba
Previous Approaches To Object Recognition

- Bags of Features (words)
- Template Matching
- Shape models

Cons

- Usually deal with fixed number of object categories
- Must retrain if adding new category
Sift Flow for Dense Scene Alignment

• Sift Flow-aligns an image with similar images

• General Idea
  • Use histogram intersection with bag of words to represent image
  • Find nearest neighbors to the image
  • Using dense sampling, match sift features between pairs of images

From Liu, et al. “Sift Flow: Dense Correspondence across Different Scenes”
Difference in sift features between corresponding pixels of images.

\[ E(w) = \sum_p \min \left( \| s_1(p) - s_2(p + w(p)) \|_1, t \right) + \]

\[ \sum_p \eta \left( |u(p)| + |v(p)| \right) + \]

\[ \sum_{(p,q) \in \varepsilon} \min \left( \alpha |u(p) - u(q)|, d \right) + \]

\[ \sum_{(p,q) \in \varepsilon} \min \left( \alpha |v(p) - v(q)|, d \right). \]

A threshold to prevent very large discrepancies from skewing the result.

Sum of magnitude of flow vector components.

Difference between flow vector of pixel \( p \) and the flow vector’s of its neighbors.

From Liu, Yuen, Torralba.
What to Take Away From the Equation

We want:

1) Small flow vectors

2) The flow vectors of the pixels surrounding pixel $p$ to all be similar

3) Pixels that are being matched to have similar sift features

**The energy function is then minimized**
Problems With Sift Flow…

- Scalability—becomes computationally expensive with large images
- Each pixel in one image can match with any pixel in the other image
- If the image size is $n \times n$, then the algorithm would be $O(n^4)$. 
Their Solution…

1) Find matching points in downsampled images
2) Proceed to the next image level and use prior knowledge gained to only search over a subset of the image
3) Repeat yet again…

*Algorithm is now $O(n^2 \log n)$ AND performs better than ordinary matching!*
Label Parsing

1) Find k-nearest neighbors to the query image using GIST matching

2) Compute SIFT flow from query image to each nearest neighbor...use this to rerank images

3) Select best candidate images

4) Use Sift flow to transfer annotations from the candidate images to the query image
\[- \log P(c|I, s, \{s_i, c_i, w_i\}) = \sum_{p} \psi(c(p); s, \{s_i'\}) \]

\[= \alpha \sum_{p} \lambda(c(p)) + \beta \sum_{\{p,q\} \in \varepsilon} \phi(c(p), c(q); I) + \log Z,\]

\[\psi(c(p) = l) = \begin{cases} \min_{i \in \Omega_p,l} ||s(p) - s_i(p + w(p))||, & \Omega_{p,l} \neq \emptyset \\ \tau, & \Omega_{p,l} = \emptyset \end{cases}\]

From Liu, Yuen, Torralba
Experiments

- LabelMe dataset
  - 2488 for training, 200 for test
  - Top 33 object categories (34th category = "other")

From Liu, Yuen, Torralba
From Liu, Yuen, Torralba
• Pixel wise recognition rate=74.75% (excluding the “unlabeled class”)

• Recognition rate for top 7 categories=82.72%

From Liu, Yuen, Torralba
More Results

Best Performance: $K=50$, $M=5$
It’s still not perfect…

From Liu, Yuen, Torralba