Style-aware Mid-level Representation for Discovering Visual Connections in Space and Time

Experiment presentation for CS3710: Visual Recognition
Presenter: Zitao Liu

University of Pittsburgh
ztliu@cs.pitt.edu

March 3, 2015
Goal: Discover connections.
**Goal:** Discover connections between recurring mid-level visual elements.
Goal: Discover connections between recurring mid-level visual elements in historic (temporal) and geographic (spatial) image collections.
Goal: Discover connections between recurring mid-level visual elements in historic (temporal) and geographic (spatial) image collections, and attempts to capture the underlying visual style.
Visual style: Appearance variations of the same visual element due to change in time or location.
The Proposed Method

Three steps:

1. Mining style-sensitive visual elements
2. Establishing correspondences
3. Training style-aware regression models
Data:

1. Three datasets are used in the paper and one dataset (CarDb) can be downloaded from the author’s website.

2. CarDb: 2.96GB with 13,473 photos. (10130 for training and 3343 for testing)

External packages:

1. VOC: extracting features.
2. LibSVM: making classification and regression.
Preprocessing:

CarDb

1920 1930 1940 1950

8 decades in total.

Zitao Liu (University of Pittsburgh)
Experiment - Step 1

Goal: **find visual elements whose appearance correlates with the “date” label.**

What code does:

1. Select “query” images from each category.
2. Random sample patches with various scales and locations from each query image.
3. Each patch is represented by a histogram of gradients (HOG).
Goal: find visual elements whose appearance correlates with the “date” (label).
What code does:

1. Find top $N (N = 50)$ nearest neighbor patches in the database.

    foreach query decade image subsets(queryDec)
    foreach query image in queryDec
    foreach match decade image subsets(matchDec)
    foreach image in matchDec(I)
    foreach detector in image I

More than 10K images.
Experiment - Step 1

Goal: find visual elements whose appearance correlates with the “date” (label).

What code does:

1. For each cluster, compute the temporal distribution of labels. (compute entropy based on year distribution)
2. Find good clusters with peaky distributions
3. Rank good clusters and select top $M$ ($M = 80$) as the discovered style-sensitive visual elements.

Peaky examples: 1920_1, 1930_1, 1930_6, 1940_2, 1960_1, 1960_2, 1960_6, 1960_12, etc.
Flat examples: 1930_9, 1930_10, 1940_6, etc.
Goal: **model the change in style of the same visual element over the entire label space.**

What code does:

1. Train a linear SVM with initial cluster images. Negative data are sampled from random Flickr images.
2. Run the learned SVM on a new subset of the data that slightly broader range in label space.
3. Keep the most confident prediction example and continue.

Cross-validation is used in each step of the incremental revision.
Good positive training examples.
Goal: **Prediction**.

What code does:

1. Train a SVM with RBF kernels on the correspondences obtained from Step 2.
2. Each training example is weighted by a transformed detection score.

Keep in mind: we have 8 decades; 10 clusters for each decades; 50 neighbors in each cluster.
Thank you!

Q & A