A Steiner tree approach to efficient object detection
Olga Russakovsky and Andrew Y. Ng  Computer Science Department, Stanford University

SUMMARY
(1) Present a classifier-agnostic approach to speeding up sliding window object detection
(2) Efficiently detect multiple object classes within a scene using a novel Steiner tree formulation for parameter selection

PROPOSING WINDOWS FOR DETECTION
Sliding window detection is inherently slow because of the large number of windows to classify ...

... especially with many object classes to detect

Goal: use segmentation to propose windows likely to contain objects to avoid exhaustive search

However:
(1) Segmentation takes time
(2) Different segmentations work for different objects

Good for mugs, bad for ski-boots

Different segmentations work for different objects

(1) Segmentation takes time
(2) Run the object classifiers only on the generated windows

Want to amortize cost of segmentation across object classes

OUTLINE
Learning:
(1) Train a binary classifier for each object class
(2) On a set of training scenes
  • Propose multiple sets of candidate windows, controlled by parameters s, k, m, b, p
  • Evaluate how good each set of windows is for detecting each object of interest
(3) Construct the Steiner graph and find the optimal parameters to use for each object class

Recognition:
(1) Segment the scene using parameters chosen for each object class, reusing computation whenever possible
(2) Run the object classifiers only on the generated windows

How to choose pipeline parameters that work well for detecting all object classes of interest?

PIPELINE FOR PROPOSING REGIONS
Unsupervised segmentation into superpixels [1]

Smoothing parameter (s):
- Original image
- Smoother image

Segmentation threshold (k):
- Small k (for detecting small objects)
- Large k (for detecting large objects)

Minimum segment size (m):
- Small m (for detecting many small segments)
- Large m (for detecting large regions)

Generating rectangular windows from superpixels

Bounding box parameter (b):
- Around each segment
- Around all adjacent pairs of segments

Trimming parameter (p):
- Low p (less trimming)
- High p (aggressively trim edges)

DIRECTED STEINER TREE FORMULATION

Given:
- A directed graph G = (E, V) with costs c(e) on the edges, a set of Steiner nodes S ⊆ V , and a root node r ∈ V ; find:
- Minimum cost tree rooted at r that spans all vertices in S.

This is NP-hard but can be efficiently approximated [2].

Claim:
- On the graph G below, a minimum cost Steiner tree corresponds to the set of parameters that minimize the overall computational cost while achieving the desired detection performance for each object class.

EXPERIMENTAL RESULTS

Comparison to sliding window, we classify 60.4x fewer windows and obtain a 10x runtime speedup while maintaining the same detection accuracy

Compared to sliding window, we classify two orders of magnitude fewer windows and obtain a 10x runtime speedup while maintaining the same detection accuracy

REFERENCES