Using the Forest to See the Trees: A Graphical Model Relating Features, Objects, and Scenes

Written by
Kevin Murphy
Antonio Torralba
William T. Freeman

Presented by
Jong Taek Lee
Joonsoo Lee

What is context?

- Any data or meta-data not directly produced by the presence of an object
  - Nearby image data
  - Scene information
  - Presence, locations of other objects

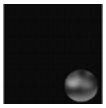
Clues for Function

- What is this?

- Now can you tell?

Low-Resolution Scenes

- What is this?



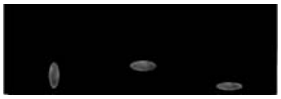
- Now can you tell?

More Low-Resolution

- What are these blobs?

More Low-Resolution

- The same pixels! (a car)



Why is context useful?

- Objects defined at least partially by function
  - Context gives clues about function
- Objects like some scenes better than others
- Many objects are used together and, thus, often appear together
  - Kettle and stove
  - Keyboard and monitor

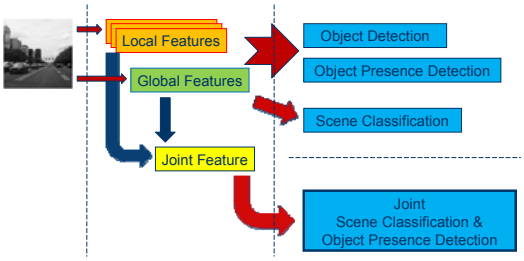
Context

- Neighbor-based context
  - Nearby image data
- Scene-based context
  - Scene information
- Object-based context
  - Presence, locations of other objects

Outline

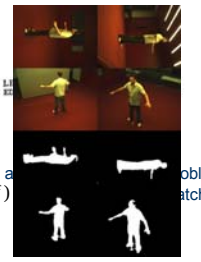
- Combining local, bottom-up information with global, top-down information using graphical model
  - Boosting-based object recognition
  - Scene learning and classification
  - Joint classification and detection

Diagram



Object detection

- Three families
  - Parts-based
  - Region-based
  - Patch-based
    - Object-detection can be a … oblem
      calculate $P(O_i^c = 1 | v_i^c)$ … atch i
      # of patches: 5000



Features for objects and scenes(1)

- Same set of features for objects and scenes
- 13 zero-mean filters $I_i(x) * g_k(x)$

- To summarize histogram : variance (k=2) and kurtosis (k=4)

$$|I_i(x) * g_k(x)|^{\gamma_k} \quad \text{for } \gamma_k \in \{2, 4\}$$

Features for objects and scenes(2)

- 30 spatial templates

(b) Dictionary of 30 spatial templates, $w(x)$.

  - Size of feature vector : 13 X 30 X 2 = 780
- Component *k* of the feature vector for patch *i*

$$f_i(k) = \sum_x \omega_k(x)(|I_i(x) * g_k(x)|^{\gamma_k})_i$$

Classifier

- The input is a set of weighted training samples (v,y,w)

$$\alpha(v) = \sum_t \alpha_t h_t(v)$$

- Regression stumps: simple but commonly used in object detection.

$$h(v) = a[v_f > \theta] + b$$

- Logistic regression

$$P(O_i^C = 1 | \alpha(v_i^C)) = \sigma(w[1\ \alpha]^T),\ \sigma(x) = 1/(1+\exp(-x))$$

Classifier - Boosting

Boosting fits the additive model

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

by minimizing the exponential loss

$$J(F) = \sum_{t=1}^{N} e^{-y_t F(x_t)}$$

Training samples

The exponential loss is a differentiable upper bound to the misclassification error.

Classifier - gentleBoosting

We chose $f_m(x)$ that minimizes the cost:

$$J(F + f_m) = \sum_{t=1}^{N} e^{-y_t(F(x_t) + f_m(x_t))}$$

Instead of doing exact optimization, gentle Boosting minimizes a Taylor approximation of the error:

$$J(F) \propto \sum_{t=1}^{N} e^{-y_t F(x_t)} (y_t - f_m(x_t))^2$$

Weights at this iteration

At each iterations we just need to solve a weighted least squares problem

Classifier

- The input is a set of weighted training samples (v,y,w)

$$\alpha(v) = \sum_t \alpha_t h_t(v)$$

- Regression stumps: simple but commonly used in object detection.

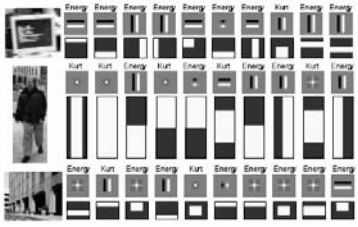
$$h(v) = a[v_f > \theta] + b$$

- Logistic regression

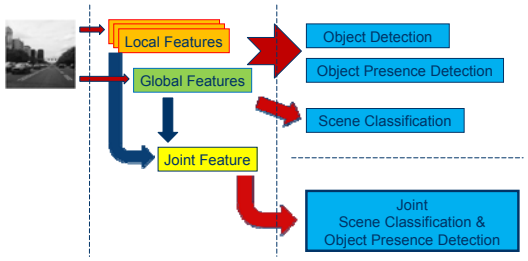
$$P(O_i^C = 1 | \alpha(v_i^C)) = \sigma(w[1\ \alpha]^T),\ \sigma(x) = 1/(1+\exp(-x))$$

Examples of learned features

- Some features after 100 rounds of boosting



Diagram



Object Localization Using Gist (1)

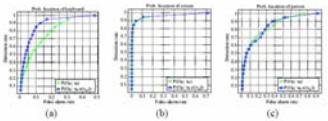
- Motivation: To improve the speed and accuracy by reducing the search space
- Approach
  - Predict probable locations/scales (continuous)
  - Take the probability into account in detectors
- How can we predict?
- **Gist: A feature vector summarizing the whole image** [A. Torralba, 2003]

Object Localization Using Gist (2)

- Calculation of the gist
  1) Take a whole image as a single patch
  2) Compute a feature vector in the same way for object detection
  3) Reduce the feature dimension by PCA (PCA-gist)
- Prediction of location/scale using the gist
  - $\hat{x}_c = E[X^c \mid v^G]$ by boosted regression
  - Cannot predict horizontal location

Object Localization Using Gist (3)

- Object Detection using gist
  1) Construct a new feature vector $f(\alpha(v_i^c), x_i^c, \hat{x}^c)$
     - Output of the boosted detector $\alpha(v_i^c)$
     - Distance between the predicted & the patch $(x_i^c, \hat{x}^c)$
  2) Train a classifier by logistic regression $P(O_i^c = 1 \mid f(\alpha(v_i^c), x_i^c, \hat{x}^c))$
- Result:

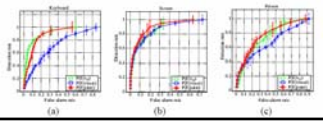


Object Presence Detection (1)

- Object presence: *is it there?*
  (*c.f.* Object localization: *where is it?*)
- Approaches (Calculate $P(E^c = 1 \mid v_{1:N}^c)$)
  - Take OR of all detectors $= \bigcup P(O^c = 1 \mid v_i^c)$
    - massive over-confidence due to the dependence
  - Approximate the OR $\approx P(O^c = 1 \mid \max \alpha(v_i^c))$
    - high probability of error
  - **Use the gist w/o object detectors**

Object Presence Detection (2)

- Presence detection using gist
  1) Train a classifier by boosting $P(E^c = 1 \mid v^G)$
  2) Construct a new feature vector $(\alpha(v^G), \alpha(v_{1:N}^c))$
     - Output of the local boosted classifier $\alpha(v_{1:N}^c)$
     - Output of the global boosted classifier $\alpha(v^G)$
  3) Combine them by logistic regression $P(E^c = 1 \mid v^G, v_{1:N}^c)$
- Result:



Object Presence Detection (3)

- Example

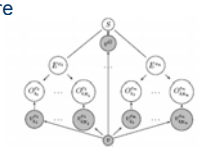
Scene Classification

- Motivation: To model the correlation among the presence of different objects
- Feature: Gist (v^G)
- Category: Office, Corridor, Street
- Approach
  1) Train a one-vs-all binary classifier by boosting
  2) Normalize the results

$$P(S=s \mid v^G) = \frac{P(S^s=1 \mid v^G)}{\sum_{s'} P(S^{s'}=1 \mid v^G)}$$

Joint Scene Classification and Object Presence Detection (1)

- Motivation: Both tasks can be done by the gist
- Approach
  - Assumption: Objects are conditionally independent given the scene
  - Graphical model: Tree structure



Joint Scene Classification and Object Presence Detection (2)

- Joint probability density model

$$P(S, E^{1:n}, O^1_{1:N}, ..., O^n_{1:N} \mid v) = \frac{1}{Z} P(S \mid v^G) \prod_c \phi(E^c, S) \prod_i P(O^c_i \mid E^c, v^c_i)$$

  - c=1:n (c; # of object classes)
  - i=1:N (N; # of patches)
  - Z; a normalizing constant
  - $P(S \mid v^G)$; output of the scene classifier
  - $\phi(E^c, S)$; table for # of occurrences of object c in scene S
  - $P(O^c_i = 1 \mid E^c = e, v^c_i) = \begin{cases} \sigma(f(\alpha(v^c_i))); & \text{if } e = 1 \\ 0 & ; \text{if } e = 0 \end{cases}$
- Train it jointly using a gradient procedure

Conclusion

- Proposed an approach to object detection using scene-based context
- Extended to basic scene classification
- Showed how to combine global and local features for object detection and scene recognition

Questions?

Reference website

- http://people.csail.mit.edu/torralba/iccv2005/slides/part_3.ppt.
- www.cs.cmu.edu/~tmalisie/presentations/opposition_to_window_scanning.ppt
- www.vision.ee.ethz.ch/~rkehl/tracking.html