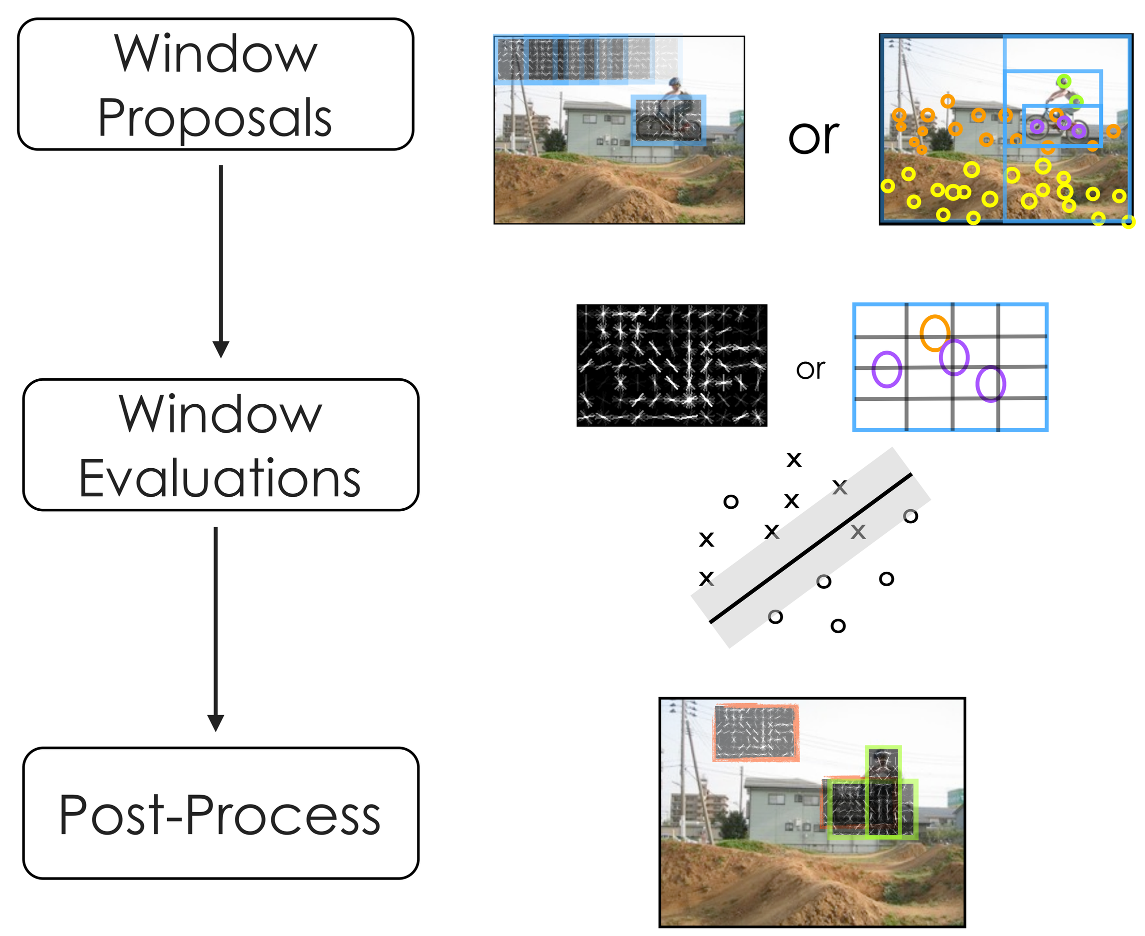
Object Detection on a Budget

Sergey Karayev, Mario Fritz, Pieter Abbeel, Trevor Darrell
Ongoing Work at UC Berkeley and MPI Informatics

Object Detection

- The most common approach to state-of-the-art object detection is through repeated classification of image windows.
- Classifier operates on a global template feature or assemblage of local features.



Efficiency and Speed

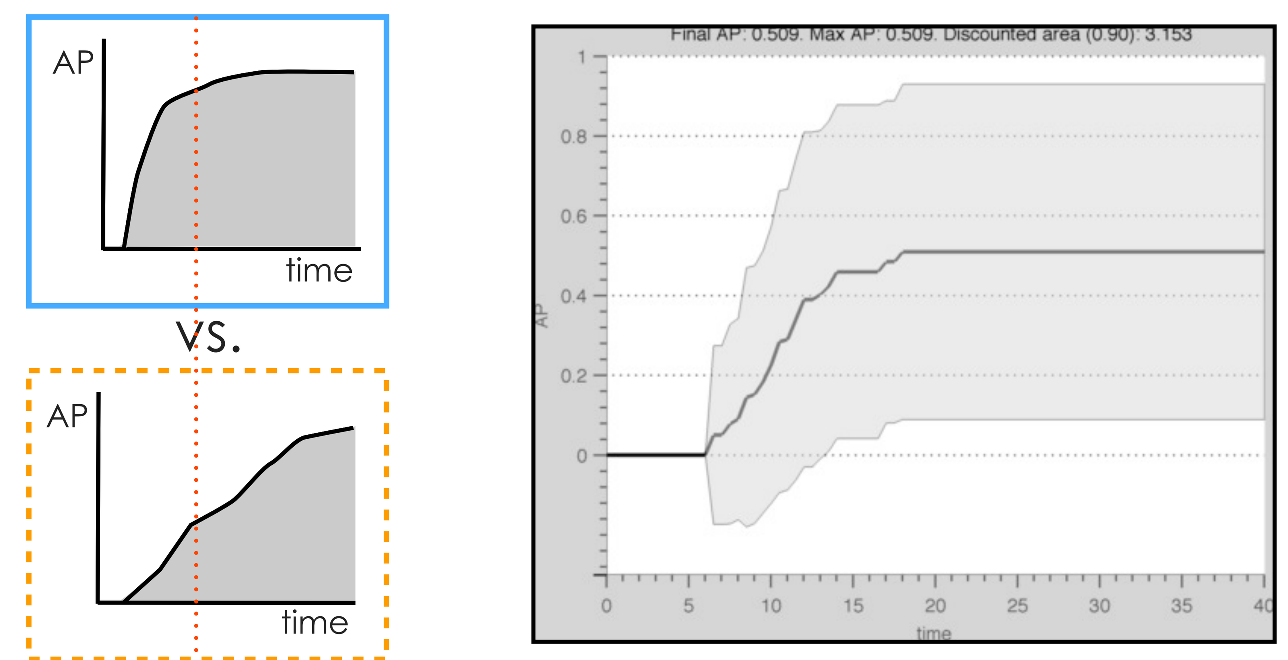
- The *window proposal* stage can be made faster with
  - a pre-filtering step that proposes only a small subset of all possible windows (e.g. "Objectness" 2010)
  - an algorithmically efficient way to pick subwindows to process (e.g. Branch & Bound 2008, Branch & Rank 2011)
- The *window evaluation* stage can be made faster with a cascaded classifier, which rejects low-likelihood windows quickly (e.g. Viola-Jones detector, 2004).
- Window proposals are key to really make a difference—fast classifiers with exhaustive windows are not as efficient.

Less windows considered →

Faster classifier ↓		
	N	log N
	1/100 N	1/100 log N

Budget: Performance vs. Time

- If we need to maintain a certain framerate, or have a large dataset to process in a fixed amount of time, we can only allot a small amount of time to each image.

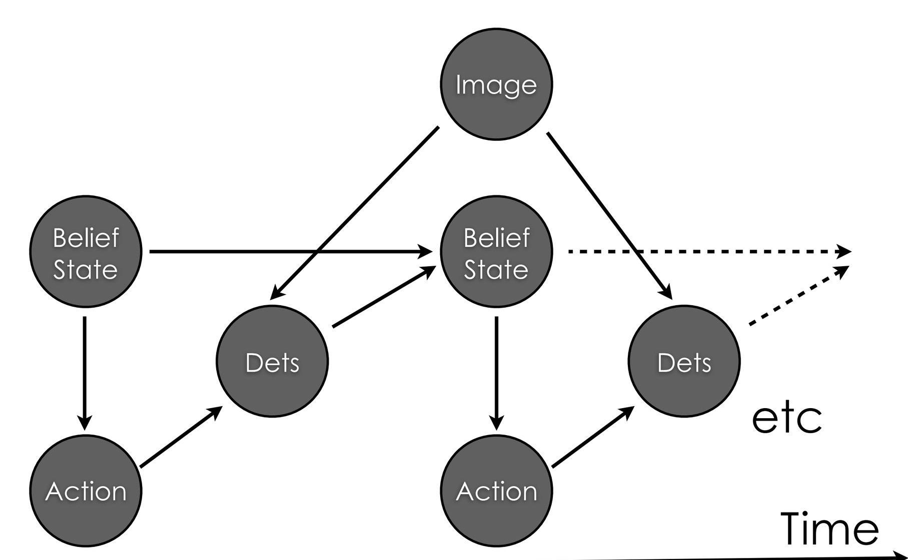


DPM Detector on PASCAL 2007

- Desired general behavior corresponds to maximizing the area under the Performance vs. Time curve: want the best performance as early as possible, after some setup time.

Our Approach

- We choose the next detection action (location and classifier) based on hypothesized contents of the image, inferred from previous detection actions.
- Window proposals follow a closed-loop policy in a partially observable decision process.



Inferring state from detections

- A detector is parametrized by two distributions $P(score|obj = true)$ and $P(score|obj = false)$.
- The likelihood of an object being in the window is inferred from these and the prior $P(obj = true)$.

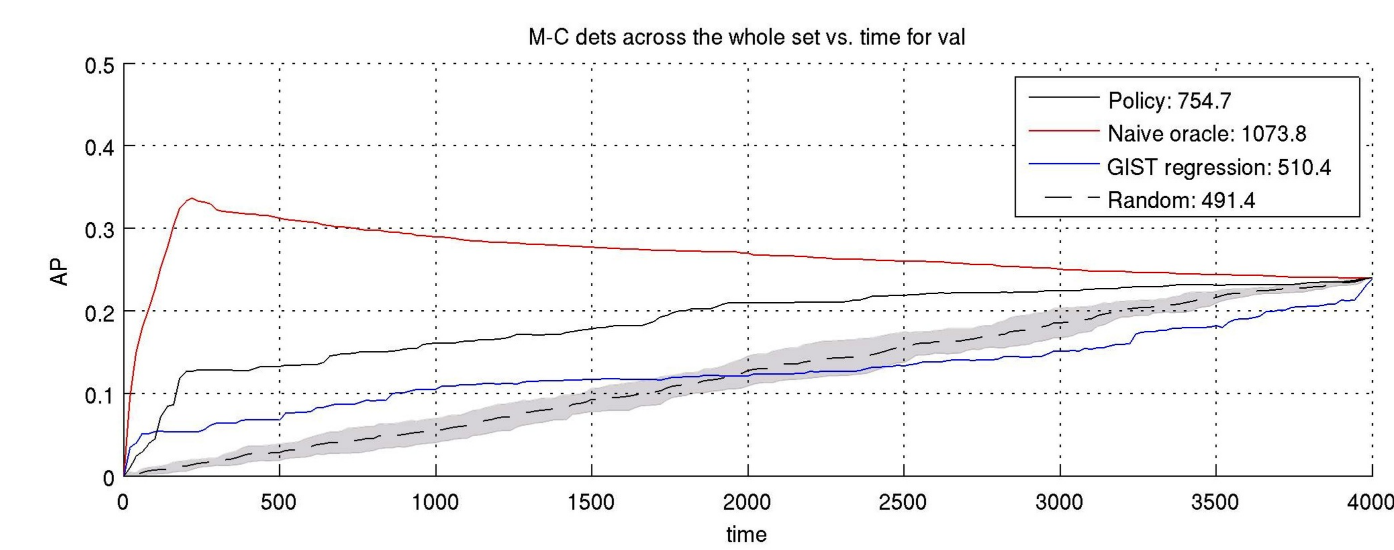
Our Approach, Continued

Selecting action

- The belief state consists of object posteriors for each class. These are treated as priors to the detection likelihoods.
- The AP can be predicted from these values, and forms our value function.

Results

- Currently evaluating state-of-the-art detectors in the AP-vs.-Time regime by modifying parameters such as stride or level of recall per cascade stage.
- Initial experiments use a simple regression from set of detections so far to the value of running a given detector.
- "Greedy" oracle deemed a good enough target, so full reinforcement learning machinery is not used.
- Although it offers clear improvement, this scheme is not able to fully derive the signal in the data.



Open Source Detection System

- Research detection systems tend to be coded from scratch, wasting efforts—state-of-the-art systems are largely alike. Additionally, research detection systems commonly rely on MATLAB, which is prohibitively expensive for individuals and small companies.
- In the course of developing this project, we aim to make our Python implementation of a modularized detection system available open-source.
- We are working with OpenCV and PCL developers to provide a unified feature computation back-end to our system.
- Please visit http://object-detection.com for more info.