Timely Object Recognition

Sergey Karayev¹, Tobias Baumgartner², Mario Fritz³, Trevor Darrell¹

¹UC Berkeley, ²RWTH Aachen, ³MPI Informatics

We evaluated on the PASCAL VOC 2007 detection task, using
• 20 Deformable Part Model detectors (one per class)
• a scene context action based on the GIST feature.

**Performance vs. Time**

Our method is compared to a random, optimal static, and oracle orderings, evaluated at different settings of start and deadline times.

**Policy Trajectories**

Action selection traces are plotted over many episodes; the size of the circles correspond to the increase in AP obtained by the action. Our policy selects actions dynamically to maximize the rewards obtained early on.

**5. Evaluation**

66% better AP than a random ordering, and 11% better than an intelligent baseline. On the timeliness measure, we obtain at least 11% better performance.

Our method is easily extensible.


**Next Steps**

• Define actions on regions of the image.

• Account for feature computation cost with multiple detector types.

**1. Motivation**

• Detect objects of many classes in an image.

• Trained detectors and classifiers already exist, but there is not enough time to run all detectors.

**Problem**

**Approach**

• Formulate timeliness evaluation of detection performance vs. time.

• Treat detectors and classifiers as “black boxes”; use reinforcement learning to find a dynamic policy for deploying them.

**Results**

• Learn to take actions that do not provide immediate reward.

• Wrapping per-class detectors in our system and setting a deadline increases the multi-class Average Precision at deadline and before.

**2. Sequential Multi-class Detection**

• Action \( a \in A \) can run detector of object class \( k \) on whole image, or classify the scene.

• The selected action returns observations \( o \): a list of detections, or an evaluated feature.

• Define the policy as taking the unacted upon action with maximum value, linearly approximated:

\[
\pi(s) = \arg \max_{a \in A} Q(s, a) = \theta^T_\pi(s, a)
\]

• Learn an accurate approximation to the expected rewards to the end of the episode:

\[
Q(s, a) = R(s, a) + \gamma \sum_{t'} \mathbb{E}[Q(s', a')]
\]

where \( R = \sum_{t=0}^{T-1} \gamma^t R(s', a') \) and \( T \) is the index of the last action before deadline time \( T_D \).

**3. Implementation**

• The final evaluation is the normalized area under the AP vs. Time curve between \( T_s \) and \( T_D \).

• Because this is additive per action, we define

\[
R(s, a) = \Delta \text{AP} \left( \frac{T_D - T_s}{T_D - T_s} \right)
\]

• Action \( a \in A \) can run detector of object class \( k \) on whole image, or classify the scene.

• The selected action returns observations \( o \): a list of detections, or an evaluated feature.

• Define the policy as taking the unacted upon action with maximum value, linearly approximated:

\[
\pi(s) = \arg \max_{a \in A} Q(s, a) = \theta^T_\pi(s, a)
\]

• Learn an accurate approximation to the expected rewards to the end of the episode:

\[
Q(s, a) = R(s, a) + \gamma \sum_{t'} \mathbb{E}[Q(s', a')]
\]

where \( R = \sum_{t=0}^{T-1} \gamma^t R(s', a') \) and \( T \) is the index of the last action before deadline time \( T_D \).

**Reward Discount**

• \( \gamma = 0 \) induces a completely greedy policy; at \( \gamma = 1 \) is full lookahead.

• \( \gamma = 0.1 \) performs best in cross-validation on the final evaluation.

**Learn \( \pi(s) \) with Monte Carlo policy iteration.**

• Gather samples under current \( \pi \) by running detection episodes on several thousand images.

• Update policy-defining weights \( \theta \) with \( L_2 \) regularization.

• Gradually decrease \( \epsilon \)-greediness of policy.

**The Belief State** posterior over class presences \( P(C|o) \) is updated with observations \( o \).

• Direct method assumes independence between classes, and simply replaces the posterior of the class(es) corresponding to the action.

• MRF method sets evidence node and runs loopy BP in a fully-connected MRF model learned with \( L_1 \)-regularization.

**4. Feature Representation and Policy Weights**

The featureization of the belief state and considered action \( \phi(s, a) \) reflects

• current probabilities of presence for all classes, and associated entropies;

• current time to \( T_s \) and to \( T_D \), and the expected time of the action.

**6. Conclusion**

If execution is stopped with only half of the detectors deployed, we obtain at least 66% better AP than a random ordering, and 11% better than an intelligent baseline. On the timeliness measure, we obtain at least 11% better performance.

Our method is easily extensible.


**Next Steps**

• Define actions on regions of the image.

• Account for feature computation cost with multiple detector types.

**AP vs Time and the Reward Function**

The timeliness evaluation is the normalized area under the AP vs. Time curve between \( T_s \) and \( T_D \).

• Because this is additive per action, we define

\[
R(s, a) = \Delta \text{AP} \left( \frac{T_D - T_s}{T_D - T_s} \right)
\]

**Reward Discount**

• \( \gamma = 0 \) induces a completely greedy policy; at \( \gamma = 1 \) is full lookahead.

• \( \gamma = 0.1 \) performs best in cross-validation on the final evaluation.

**Learn \( \pi(s) \) with Monte Carlo policy iteration.**

• Gather samples under current \( \pi \) by running detection episodes on several thousand images.

• Update policy-defining weights \( \theta \) with \( L_2 \) regularization.

• Gradually decrease \( \epsilon \)-greediness of policy.

**The Belief State** posterior over class presences \( P(C|o) \) is updated with observations \( o \).

• Direct method assumes independence between classes, and simply replaces the posterior of the class(es) corresponding to the action.

• MRF method sets evidence node and runs loopy BP in a fully-connected MRF model learned with \( L_1 \)-regularization.

**4. Feature Representation and Policy Weights**

The featureization of the belief state and considered action \( \phi(s, a) \) reflects

• current probabilities of presence for all classes, and associated entropies;

• current time to \( T_s \) and to \( T_D \), and the expected time of the action.

**6. Conclusion**

If execution is stopped with only half of the detectors deployed, we obtain at least 66% better AP than a random ordering, and 11% better than an intelligent baseline. On the timeliness measure, we obtain at least 11% better performance.

Our method is easily extensible.


**Next Steps**

• Define actions on regions of the image.

• Account for feature computation cost with multiple detector types.