Contextual vision

Antonio Torralba, Kevin Murphy, William T. Freeman, Leslie Kaelbling

We are interested in detecting objects, such as cars and faces, in images. Standard approaches look at each patch of the image in isolation and attempt to classify it as positive (containing the object) or negative (part of the background). However, ignoring the overall image can lead to ambiguous or erroneous results: see Figure 1 for an example.

Figure 1: The power of context: depending on the overall image, we interpret an object as **(left)** a car or **(right)** an ash tray, even though the blob inside the ellipse are identical. From [2].

Following on from Torralba [2], we define global image features, which capture the "gist" of the image in a compact way. We combine the global features with the local features, used by standard object detectors, in two distinct ways. First, we can use the global features to classify the type of scene (e.g., office or street scene). This can be used to predict if the object is present in the image, without running any detectors. In earlier work [3], we used these global features in conjunction with a hidden Markov model (HMM) to perform online scene categorization and place identification, as we walked around the Tech Square environment (outdoors and indoors) with a wearable camera.

Having inferred the scene type and decided which objects are likely to be present, we can use the global features to predict the location and scale of each object class. Finally, we can run a standard object detector inside the primed image region. This is much more efficient than the standard approach of running the detectors for all the object classes and for all locations/ scales, before even looking at the image.

The various pieces of the system are glued together using a probabilistic graphical model, which can be trained using standard techniques [1]. Figure 2 gives a snapshot of part of the system in action. It shows the system detecting keyboards and cars, using global features alone, i.e., before running the object detectors. The image on the left shows a typical input image; it has been classified as an office scene. The middle image highlights the area where the system expects to find a keyboard (based on what the system has learned from other images with similar global features); this is where we will apply our keyboard detector. The right image is dark, indicating that the system doesn't expect to find a car in this kind of image.

The circular nodes in Figure 2 represent random variables in the model. The shaded v_g node represents the gist of the image, which is observed (known); the states of the remaining unshaded nodes have to be estimated. The $Scene$ node represents the scene-type (here, office). $E^{kbd} = 1$ represents the presence of a keyboard, and $E^{car} = 0$ represents the absence of car. L^{kbd} represents the expected location of the keyboard (shown in the middle figure); L^{car} is the expected location of the car, which is ignored (since $E^{car} = 0$). Finally, $O_i^{kbd} = 1/0$ means that a keyboard is present/absent at location i within this image, and similarly for O_i^{car}.

We are showing the system in an intermediate state, where its estimate of the values of the various nodes is conditioned on the global features only. The final step (not shown) is to run the keyboard detector (which uses local image features) in the highlighted region. This gives us a better estimate of

whether the object is present, and if so, its location (i.e., we compute $P(E^{car}|v_g, v_l)$ and $P(O_i^{car}|v_g, v_l)$, where v_g are the global features and v_l the local features, and similarly for keyboards). This information can then be passed back up to the scene-type node, and used as a prior for the next frame in the sequence. The resulting system is able to detect objects far more quickly and robustly than a method that just uses local features and no graphical model.

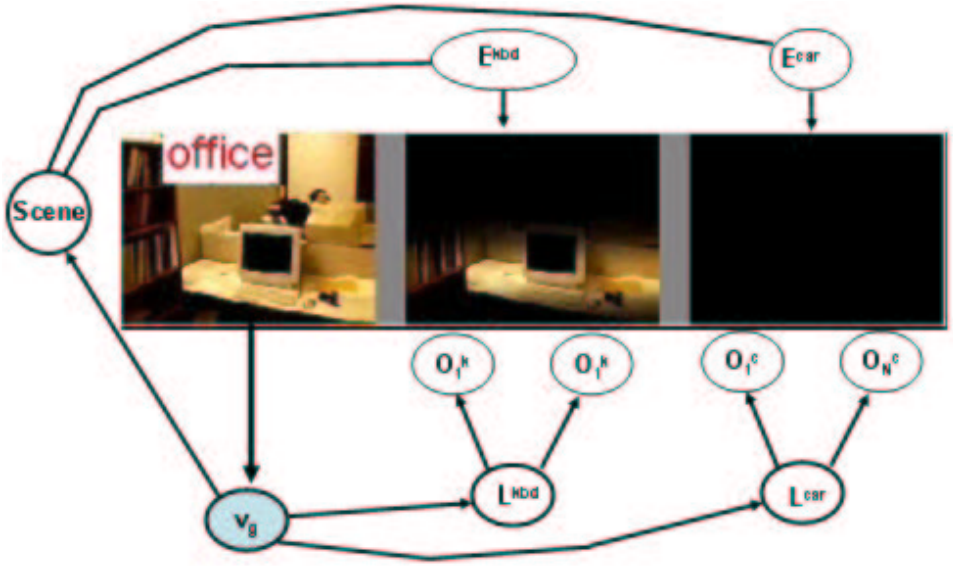

Figure 2: Snapshot of part of the system. The image on the left shows a typical input image; it has been classified as an office scene. The middle image highlights the area where the system expects to find a keyboard (given this image); this is where we will apply our detector. The right image is dark, indicating that the system doesn't expect to find a car in this kind of image. The circles represent random variables inside the model, which is used to perform probabilistic inference about the type of scene, and the presence/ location of different types of objects.

Research Support: This research was supported in part by the Nippon Telegraph and Telephone Corporation as part of the NTT/MIT Collaboration agreement, and in part by DARPA contract #DABT63-99-1-0012.

References:

[1] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *Neural Info. Processing Systems*, 2003.

[2] A. Torralba. Contextual priming for object detection. *Intl. J. Computer Vision*, 53(2):153–167, 2003.

[3] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition. In *Intl. Conf. Computer Vision*, 2003.