Learning to Recognize Objects in Images Using Anisotropic Nonparametric Kernels

Douglas SUMMERS-STAY and Yiannis ALOIMONOS
University of Maryland, College Park

Abstract. We present a system that makes use of image context to perform pixel-level segmentation for many object classes simultaneously. The system finds approximate nearest neighbors from the training set for a (biologically plausible) feature patch surrounding each pixel. It then uses locally adaptive anisotropic Gaussian kernels to find the shape of the class manifolds embedded in the high-dimensional space of the feature patches, in order to find the most likely label for the pixel. An iterative technique allows the system to make use of scene context information to refine its classification. Like humans, the system is able to quickly make use of new information without going through a lengthy training phase. The system provides insight into a possible mechanism for infants to quickly learn to recognize all of the classes they are presented with simultaneously, rather than having to be trained explicitly on a few classes like standard image classification algorithms.

Keywords. Object Recognition, Anisotropic, Non-Parametric.

Introduction

When we look at the world, we are able to classify many things within the field of view quickly, simultaneously, and effortlessly. Most models of attention assume that when we first look at a scene, the brain pulls out only simple features such as contrast, information density, saturation, creating what is known as a “saliency map.” These features are thought to provide cues for where to fixate in an image, and that objects are only recognized when they are the center of attention indicated by the direction of gaze.

A few recent works have shown a situation more complicated and interesting. The earlier experiments in this area simply asked participants to look at a scene and describe what they saw. In this case, the location of fixations was predicted pretty well by the low-level features described above. When participants are asked instead to look for a particular item in a natural image, however, the first few fixations were better predicted by the location of the object to be found [1]. This seems to indicate that from the first glance at a scene, before the brain would have time to do anything that requires anything as slow as conscious reasoning or fitting of a complex model, it is already able to classify many objects in a scene correctly and in parallel. Only after this process is *completed* do we fixate on the object of interest in order to begin these slower and more accurate processes which require a focused attention.

Our system is an attempt to model this aspect of pre-attentive vision. In brief, we

- collect biologically plausible rich features, called “prototypes” from training images with known labels

- use these to classify all features on these same images
- learn a multi-layer model that can refine these estimates using context
- apply the multi-layer model to test images

1. Object Recognition in the Brain

The following is a sketch of the current consensus about the process of object recognition in primates. The data from the eyes is streamed along the ventral visual pathway beginning in the primary visual cortex (V1) and ending in the inferotemporal cortex (IT). This in turn informs the prefrontal cortex, where the information can be used for taking action. The entire process from V1 to IT only takes about 30 ms in humans. [2] (Information about location in the image also begins in V1 but follows a different path. We do not attempt to imitate this behavior in our model.)

The first cells along the pathway, the simple (S1) cells, are similar to local Gabor filters at a particular orientation and scale. Complex (C1) cells integrate the information from a small number of these S1 cells, responding to oriented edges over a wider range of locations and scales. The input of multiple C1 cells, in turn, are used to create more and more complex filters that respond to particular arrangements of multiple edges over larger and larger areas of the image (S2 and C2 cells.) [3] Cells at the end of this process act like radial basis functions, responding strongly to image regions that contain the pattern of interest, and falling off in Gaussian fashion as the similarity between the input patch and the prototype decreases. [4]

Up to this point the process is largely feed forward. But within the inferotemporal cortex, these prototypes receive feedback from the prefrontal cortex [5], influencing the interpretation of inputs so that ambiguous areas are resolved into familiar objects through association with the immediate context. For example, a distant brown blob might be interpreted as a shoe if it is found at the bottom of a leg, or as hair if found at the top of a head.

For some cells in the IT cortex, the visual similarity between inputs is less important than semantic similarity. Cells that respond strongly to frontal views of faces, for example, respond partially to profiles of faces, even though their appearance is not similar. [6]

2. Object Recognition by Our System

Our system follows this natural model closely for the first stages of processing, approximating the action of S1, C1, S2, and C2 cells. (This part of the system uses a variation on the HMAX features described in [7].) Randomly selsected 64 x 64 patches of the training images are fed into this software, and the results are 256 dimensional vectors which encode much of the shape information in the patches in a compact way. These vectors (which we will call ‘prototypes’) are associated with training labels, giving the classification of the object at the center of the patch.

A sliding window is applied, and the approximate nearest neighbors to each windowed region from among these prototypes are returned. What has been described so far is similar to [8]. We extend the model beyond this with multiple layers of prototypes that do not merely classify an image as a whole, but create a classification map that shows which regions of the image belong to which class.

For each sampled point in the images, we find the most similar prototypes and average them, making use of a rich weighting scheme (discussed later.) Using the correct label maps for these training images, the system learns what it ought to produce when a particular pattern of label maps is generated. We do this by creating a new set of prototypes in a second layer, which take as input not just a patch of the original image, but also the associated patch from the estimated label map created by the first layer. This process can be repeated several times.

When testing images are presented, the exact same process is followed, except that new prototypes are not collected. Instead, each layer of prototypes create during training is applied in sequence, making use of the estimated label map generated by the previous layer.

2.1. Training

```
1.  A set of training images are collected.
2.  Corresponding label maps are created.
3.  For each layer,
4.      Features are collected at many random locations within these training pairs.
5.      An index is created to enable fast searching among these features.
6.      For each training image,
7.          A feature is collected at each pixel in the image.
8.          A set of similar features are found.
9.          A weighted average of the labels of these features is found.
10.         An estimated label map is created from these labels.
```

2.2. Testing

```
1.  For each test image
2.      For each layer,
3.          Follow steps 7-10 above.
```

Though we have used biological language to describe the process in this paper, the problem can also be formulated as a straightforward statistical inference, as described in [9]. Let a training image be represented by the vector $X = (x_1, ..., x_n)$. Each of the x_i represents a single pixel. Each training image comes with a corresponding ground truth map $Y = (y_1, ..., y_n)$ where $y_i \in \{1..K\}$ is the label for each pixel i, and an estimated probability of detection map $W = (w_1, ..., w_n)$ where all the w_i are initially set to the same value. We would like to learn to estimate $p(y_i | X \text{ and } W)$. Since this is too large a space to attempt to learn directly (a megapixel image would result in a million dimensional space), we instead learn $p(y_i | V \subset (X \text{ and } W))$, where V is a subset of X and W consisting of a patch of pixels surrounding x_i. and a patch surrounding w_i.

Once we have learned $p(y_i | V)$, we apply it to the patch surrounding each pixel x_i in each training image X. In this way, we create an estimated label map W for each of the training images. In this map W, some pixels will be correctly labeled while their neighbors are incorrectly labeled. Since we have the truth map Y for each image, we can learn, for example, that a pixel w_i surrounded by pixels belonging to a particular class K is more likely to itself belong to that class. Moreover, by using both the estimated map W and the original image X together as one half of the training pair, we can do a better job of estimating y_i than if we only had the original image X. This process of iteratively creating new estimated detection maps continues until the maps no longer improve.

The sharing of context information between neighboring pixels introduced in this way is comparable to how belief propagation networks or conditional random fields (CRFs) have probabilities defined for sharing probabilities between neighbors.

3. Anisotropic Interpolation

While the prototypes are a compressed representation of the patches they are derived from (a 64 x 64 patch with 4096 pixels is represented by only 256 values) they are still too high dimensional for approximate nearest neighbor algorithms to work well. The 100 nearest neighbors will contain some correct matches but also many incorrect matches. The usual way to weight the neighbors is with a Gaussian function on the distance from the point to be estimated. Unfortunately, in high dimensional spaces, all points are approximately the same distance apart. This is one aspect of the 'curse of dimensionality.' However, the relevant data lies on a lower dimensional manifold embedded in this 256 dimensional space. Because of this, using adaptive anisotropic kernels gives a substantial improvement over the standard isotropic Gaussians.

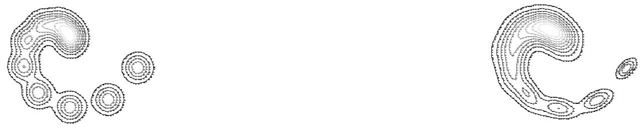

Figure 1. Isotropic Gaussian kernels (left) and anisotropic Gaussian kernels (right) on the same ten points.

The advantage can be seen in the following illustration. Ten points forming an expanding spiral. The points represent prototypes. The spiral is 2-dimensional for illustrative purposes—the actual prototypes are points in a 256 dimensional space. In the first illustration, the weights of each prototype are given by an isotropic Gaussian function. When the prototypes are very similar, the points are close together, and the interpolation between them is reasonably accurate. However, when they are widely spaced, each prototype lies in its own island. Test features which are very similar to one particular prototype will be classified correctly, but ones that lie halfway between two prototypes will not be.

In the second illustration, anisotropic kernels are used. These are elongated in the direction of neighboring points of the same class. In this case, the points form a nearly connected spiral, correctly estimating the shape of the underlying manifold. This effect is even more pronounced in higher dimensional spaces where the weight is concentrated in one direction among hundreds, rather than one direction out of two in the illustration.

The methods we used to estimate the shape of these kernels is not biologically plausible, relying on taking the inverse of a covariance matrix. (See [10] for details and formulae for these anisotropic kernels.) The shape of these kernels may be formed by interaction among similar prototypes gradually "reaching out" towards their neighbors in the same process that allows redundant prototypes to be gradually eliminated in the learning process. This, however, is purely speculative at present.

4. Results

We tested the application on the Weizmann horse database [11]. This database has large variations in the appearance, lighting, and pose of the horses and variations in background appearance. The system was trained on 300 of the images and tested on the remaining 27 (See Figure 2.) 500,000 prototypes were collected at random from the training images for each of the five layers. The system used 64 x 64 patches, and created 256 dimensional prototype vectors.

The system was able to not only detect the presence of horses, but correctly segment many of the limbs in 24 of the 27 images. Detection is made a little easier by the fact that each image contains only one horse, and there are no partial occlusions. However, due to the windowed nature of the detection algorithm, these factors would not be expected to be very problematic for this system. In addition, the horses are all from roughly the same angle. This means fewer prototypes are needed to learn the class than would otherwise be the case.

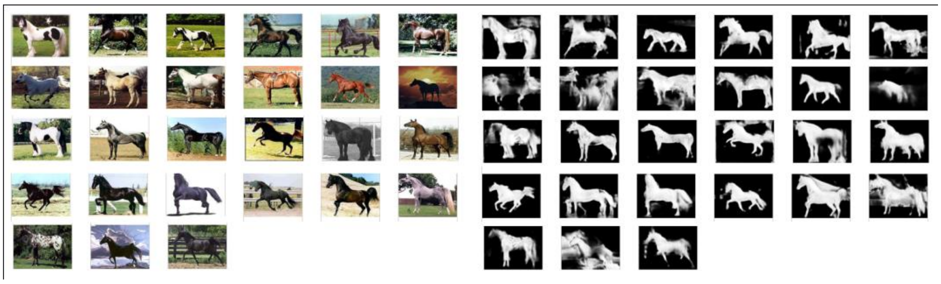

Figure 2. Test set. Images (left) and corresponding detection maps (right).

5. Conclusion and Future Directions

This seems to be a promising approach to forming rough segmentations of the classes of objects in a scene prior to fixation and segmentation. We have begun experiments on including stereo and motion information, to learn to recognize 3D objects and motions as well as image classes.

An advantage of this system is that it requires no more resources to learn many classes from a set of training images than it does to learn just two from the same set. Even classes not explicitly specified, such as head or limb detectors in the case of the horse database, are recognized as being visually and semantically similar implicitly. Labeling a single horse leg, for example, could bring up a cluster of similar horse legs because all would activate the same prototypes. In this way, the system is learning something about everything in the training images, even when it doesn't have a name for the groups it recognizes as similar. In this way it could combine supervised with unsupervised learning.

One other interesting possibility is to replace the mapping to discrete labels with a mapping into some kind of semantic space. Objects recognized as being semantically associated would be able to influence the classification of nearby objects in the scene (the presence of a spoon and plate might help to resolve an ambiguous detection as a cup.)

References

[1] W. Einhäuser, M. Spain, and P. Perona, Objects predict fixations better than early saliency. Journal of Vision, 8(14):18, 1–26, 2008

[2] JJ Foxe, GV Simpson. Flow of Activation from V1 to frontal cortex in humans. Experimental Brain Research, 2002.

[3] J Mutch, DG Lowe. Multiclass object recognition with sparse, localized features. CVPR 2006

[4] T. Serre, L. Wolf, T. Poggio. Object recognition with features inspired by visual cortex. CVPR 2005.

[5] EK Miller, CA Erickson, R Desimone. Neural mechanisms of visual working memory in prefrontal cortex of the macaque. Journal of Neuroscience, 1996.

[6] R Desimone, TD Albright, CG Gross. Stimulus selective properties of inferior temporal neurons in the macaque. Journal of Neuroscience, Vol 4, 1984.

[7] M Reisenhuber, T Poggio. Heirarchial models of object recognition in cortex. Nature Neuroscience 2, 1999.

[8] M Reisenhuber, T Poggio. Heirarchial models of object recognition in cortex. Nature Neuroscience 2, 1999.

[9] Tu, Zhuowen. Auto-context and Its Application to High-level Vision Tasks. Proc. of IEEE Computer Vision and Pattern Recognition (CVPR), 2008.

[10] Thomas Brox, Bodo Rosenhahn, Daniel Cremers and Hans-Peter Seidel. Nonparametric Density Estimation with Adaptive, Anisotropic Kernels for Human Motion Tracking. Lecture Notes in Computer Science, 2007.

[11] Weizmann horse database can be found at http://www.msri.org/people/members/eranb/

[12] R. Haralick, K. Shanmugam, and I. Dinstein. Texture Features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics, 3(6), 1973.

[13] H. Seo, and P. Milanfar. Training-free, Generic Object Detection using Locally Adaptive Regression Kernels. Accepted for publication in IEEE Trans. on Pattern Analysis and Machine Intelligence, June 2009

[14] Wu, B., & Nevatia, R.. Detection and Segmentation of Multiple, Partially Occluded Objects by Grouping, Merging, Assigning Part Detection Responses. Int J Comput Vis (2009) 82: 185–204

[15] L. Zhao and L. S. Davis. Closely Coupled Object Detection and Segmentation. ICCV, 2005.

[16] J.Winn and J. Shotton. The Layout Consistent Random Field for Recognizing and Segmenting Partially Occluded Objects. CVPR, 2006.

[17] Wu, B., & Nevatia, R.. Detection and Segmentation of Multiple, Partially Occluded Objects by Grouping, Merging, Assigning Part Detection Responses. Int J Comput Vis (2009) 82: 185–204

[12] A. Hollingworth and J.M. Henderson, Accurate visual memory for previously attended objects in natural scenes. Journal of Experimental Psychology: Human Perception and Performance, 28: 113-136, 2002.

[13] A. Hollingworth, Constructing visual representations of natural scenes: The roles of short- and long-term visual memory. Journal of Experimental Psychology: Human Perception and Performance, 30: 519-537, 2004.

[14] R.A. Rensink, The dynamic representation of scenes. Visual Cognition, 7:17-42, 2000.