Models for generic visual object detection

Kevin Murphy

11 May 2005

In this document, we provide a concise summary of various models that have recently been proposed for detecting and localizing categories of objects in clutter. We also discuss related problems such as class presence detection (image classification), and pixel/ patch/ region labeling. Wherever possible, we use graphical models, either probabilistic or deterministic, as a representation.

Notation (for reference): upper case letters denote unknown quantities (random variables), lower case letters denote constants. $n \sim 5000$ is the number of patches, $d \sim 500$ is the number of interest points, $m \sim 5$ is number of parts per object, N is the number of objects (unknown), v is the image, $v_{i=1:n}$ is a (representation of) the i'th image patch, $H_{i=1:n} \in \{1, \ldots, \ell\}$ is a label for patch i, $X_{i=1:N} \in \{1, \ldots, n\}$ is the location of an object, $Y_{i=1:m} \in \{1, \ldots, n\}$ is the location of object part i, $l_{i=1:d}$ is the location of the i'th interest point, $a_{i=1:d}$ is the appearance of the i'th interest point, $C_{i=1:d} \in \{1, \ldots, c\}$ is the cluster label for i'th interest point.

1 Sliding window classifiers

Sliding window classifiers were first proposed by [RBK95] (if not earlier), based on feedforward neural nets. Since then various other kinds of classifiers have been explored, such as SVMs [PP00], boosted decision stumps [VJ04], convolutional neural nets [LHB04], etc.

Each window or patch of the image v centered at location i, denoted v_i, is transformed into a feature vector and then classified into foreground object (1) or background (0), where $i = 1 : n$, where n is the number of patches. Let H_i denote the unknown label for patch i. The classifier produces a score $s_i = f(v_i)$ which can be converted into a probability e.g., using a sigmoid transform [Pla99]: $P(H_i = 1|v_i) = \sigma(a\sigma_i + b)$ (where parameters a and b are estimated on a validation set to avoid overconfidence). Each patch is classified independently: see Figure 1.

Of course, the ultimate desired output is not one binary label per patch, but rather, the number and location(s) of the objects. (We ignore scale for simplicity.) The standard approach is to threshold $P(H_i = 1|v_i)$ (or equivalently the score σ_i), and/or to apply non maximal suppression, and then to report athe number N and locations $X_{i=1:N} \in \{1, \ldots, n\}$ of all the local maxima.

Note that although the patch classifier is learned, the nonmaxsup procedure is typically designed by hand (even though it has several tunable constants). One reason for this is that the size of the output from the nonmaxsup "box" is variable — it is not a fixed-length feature vector, making most learning methods inapplicable.

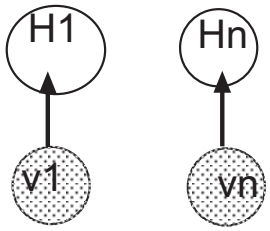

Figure 1: Patch classifier as a degenerate Bayes net. Shaded nodes are observed, unshaded are hidden. v_i is the (feature vector for the) i'th image patch; $H_i \in \{0, 1\}$ is the label for the patch. $n \sim 5000$ is the number of patches.

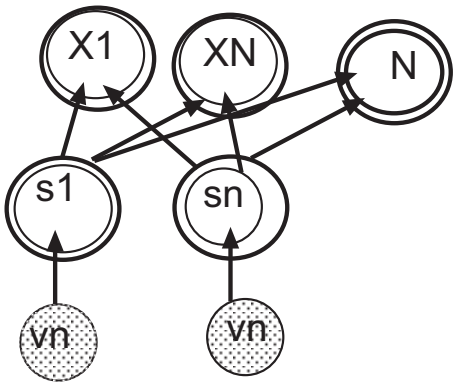

Figure 2: Sliding window classifier as a feedforward circuit. v_i is the feature vector for the i'th patch. s_i is the score of the classifier on this patch. n is the number of patches. N is the estimated number of objects, and $X_{1:N}$ are their estimated locations. Typically these estimates are computed by applying nonmaximal suppression to the matrix of scores $s_{1:n}$. Double rings represent deterministic nodes. However, this is not really a Bayes net.

We can represent the whole system pictorially as in Figure 2. However, this is not a Bayes net (or any indeed, a probabilistic model of any kind), for several (interesting) reasons:

- The size of the graph (number of X nodes) is not fixed a priori, but is a deterministic function of all the s_i nodes.
- The score nodes are deterministic functions of their input, $s_i = f(v_i)$, not random variables.
- In a Bayes net, each conditional probability distribution has to be a mapping from values to distributions, not from distributions to distributions. eg we have to write $P(X|H_{1:n})$, not $P(X|\{P(H_i)\})$.

2 Parts-based sliding window classifiers

It has recently become popular to break the problem of classifying a window as face or not (or whatever class) into stages, where one first applies a sliding window classifier plus local maximum finder within the window for various parts or components, such as eyes, nose, mouth; then one treats the scores and locations of these detections as a fixed length feature vector representing the original window, and classifies that (see eg. [MPP01]). The second layer classifier can test for the geometrical consistency of the found part detections within the window. Since the system is not a probabilistic model, it cannot "hallucinate" the location of missing parts, based on top down information. Also, it requires manually labeled parts. On the other hand, it is simple and fast.

3 Random field models

Grid-structured Markov random fields (see Figure 3) have been widely used for low-level vision tasks, such as stereo, optical flow, etc. An MRF is a generative model:

$$P(H_{1:n}, v_{1:n}) = \prod_i P(v_i|H_i) \left[\frac{1}{Z} \prod_{<ij>} \psi_{ij}(H_i, H_j) \right]$$

where $H_i \in \{1, \ldots, \ell\}$ is a discrete random variable representing e.g., quantized depth of the pixel. $P(v_i|H_i)$ is the probability of generating local patch v_i given label H_i, and $\psi_{ij}(H_i, H_j)$ is a compatibility potential between neighboring states. Note: exact inference in these models (computing $P(H_i|v)$) is intractable, so it is common to use loopy belief propagation [FH04], graph cuts [BVZ01], Gibbs sampling [GG84], Swendsen Wang [BZ04], etc.

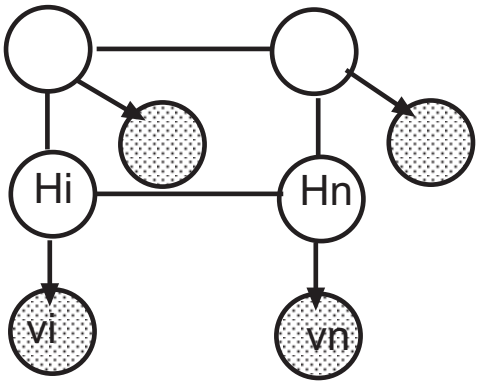

Figure 3: A small grid Markov random field (MRF). Here hidden nodes are $H_i \in \{1, \ldots, \ell\}$. We have an undirected hidden backbone; each hidden node generates its local evidence. (Hence, strictly speaking, this is a chain graph [Bun95], since it combines directed and undirected edges.)

Since generative models make strong assumptions about the data generating mechanism, a better alternative is to use a conditional random field (CRF),

$$P(H_{1:n}|v) = \frac{1}{Z(v)} \prod_i \phi_i(H_i|v) \prod_{<ij>} \psi_{ij}(H_i, H_j|v)$$

where the notation $\psi_{ij}(H_i, H_j|v)$ makes explicit the fact that the potentials can depend on the entire image v [LMP01, KH03]. For example, we can require that neighboring labels be similar, except when an edge is observed to fall between them. Also $\phi_i(H_i|v)$ is now an arbitrary potential, e.g., it can be the output of a discriminative classifier. Inference in a CRF has the same complexity as in an MRF, but learning is even harder (since now the local evidence is no longer causal).

Grid-structured MRFs/CRFs are usually used for low level vision (eg stereo or optical flow), but they have also been used for binary pixel/patch labeling (e.g., H_i = building-like-texture or not [KH03]), and multi-class pixel/patch labeling [TMF04, HZCP04]. Note that the output of these systems is a fixed size set of patch labels, not a variable-sized set of objects.

For higher level vision, it is helpful to escape the tryany of the 2D grid. One approach is to make the nodes in the graph might be regions (superpixels) output by a segmentation algorithm, such as Ncuts, and the connections are to nearest neighbors; the goal is to classify each region with a semantic label, such as sky, water, etc [CdFB04]. Alternatively, the nodes might be located at interest points, and the goal is to label the nodes with semantic object parts [KH04]. Carbonetto [CS05] combines superpixels and interest points in a CRF, trained in a semi-supervised way. Again, however, the output of these systems is a fixed size set of patch labels, not a variable-sized set of objects.

4 Pictorial structures

A more probabilistic approach to parts-based object detection is to build an CRF over the parts [FH05]: see Figure 4. Here each $Y_i \in \{1, \ldots, n\}$ is a discrete variable representing the location of body part i. (The fact that this is a discrete random variable is important, since it allows an easy way to handle multi-modality.) The local evidence $\phi_i(Y_i = x)$ could be the outputs of a sliding window part detector, as explained above. Implicitly on each edge is a $\psi_{ij}(Y_i = x, Y_j = x')$ factor, which represents spatial compatibilities between the two part locations (c.f. Figure 5).

Note: here m is the number of pre-specified parts, not the number of objects: there is assumed to be exactly one object present. If there are multiple instances of the object class, one can in principle enumerate the N-best hypotheses, although these modes are not usually spatially distinct, but rather minor configurational differences. Hence other techniques (such as sampling) must be employed for the multi object case.

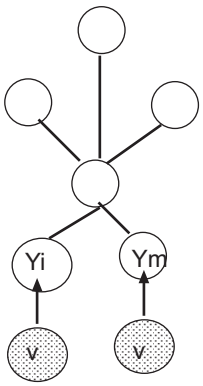

Figure 4: A CRF for detecting the location of m body parts, $Y_i \in \{1, \ldots, n\}$. The fact that the local evidence for each node Y_i is potentially a function of the whole image is indicated by drawing an arrow from v to each Y_i.

Exact inference (using belief propagation) in this model takes $O(mn^{w+1})$, where w is the treewidth of the graph. For a tree-structured model, $w = 1$, leading to the well-known $O(mn^2)$ complexity. Since $n \sim 5000$ is the number of patches examined by the local evidence, this can be expensive. In the case that ψ_{ij} has a special kind of sparse structure (e.g., arising from discretizing a Gaussian), "fast methods" can be used to reduce the complexity to $O(mn)$ [FH05],[Nando].

5 Constellation model

A very influential model for object class detection is the constellation model [FPZ03]. This is a generative model for interest point locations and appearances. That is, we pre-process the image with an interest point detector, that returns a small number d (typically 500) set of image locations l_i, called keypoints. These are often annotated with a local scale estimate σ_i. One can then derive a descriptor of l_i, such as a 128 dimensional SIFT vector [Low04], or one can just take an image patch centered at l_i. We will denote the appearance of local region i as $a_i = f(v, l_i, \sigma_i)$. Now the image has been reduced to a labeled bag of points.

The constellation model defines

$$P(a_{1:d}, l_{1:d}) = \sum_{\pi} P(a_{1:d}, l_{1:d}|\pi)P(\pi)$$

where π is an assignment of object parts to image detections (if the part is observed); unassigned detections are assumed to be caused by the background. Given an assignment, we can write

$$P(a_{1:d}, l_{1:d}|\pi) = P_{fg}(l_{\pi_1}, \ldots, l_{\pi_m}) \prod_{i=1}^{m} P_{fg}(a_{\pi_i}) \prod_{j \notin \pi} P_{bg}(a_j) P_{bg}(l_j)$$

The appearance model assumes a_i is a PCA reduction of the image patch v_i. The spatial model, $P(Y_{1:m})$, is a full covariance Gaussian (corresponding to a fully connected undirected graphical model: see Figure 5).

The constellation model can be represented as a Bayesian multinet [Bil00], but this is not very intuitive: essentially the only uncertain random variable is π. Conditioned on π, the locations and apperances are determined by the data! For example, suppose we have $m = 3$ parts and $d = 4$ interest points, and π assigns part 1 to observation 2, part 2 is not observed, and part 3 is assigned to observation 1 (so observations 3 and 4 are caused by the background). The likelihood of this value of π can be determined by evaluating the fully observed graphical model in Figure 6(a). The prior for π comes from a Poisson event model (borrowed from the radar tracking community [BSF88].

One can perform localization of a single object by evaluating all $O(m^d)$ assignments (values of π) in this way.[1]

[1] Note that one can use A* search instead of exhaustive enumeration, but this cannot be applied during learning, when the model parameters are uncertain. This has motivated a sequential learning strategy [HL04].

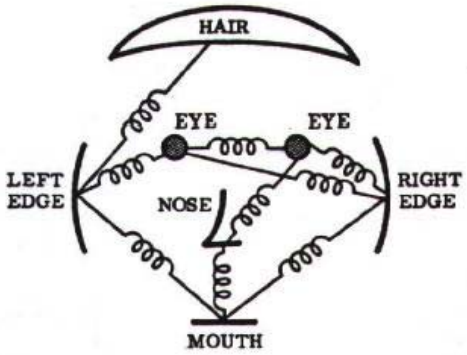

Figure 5: Gaussian spatial model of face parts, from [FE73].

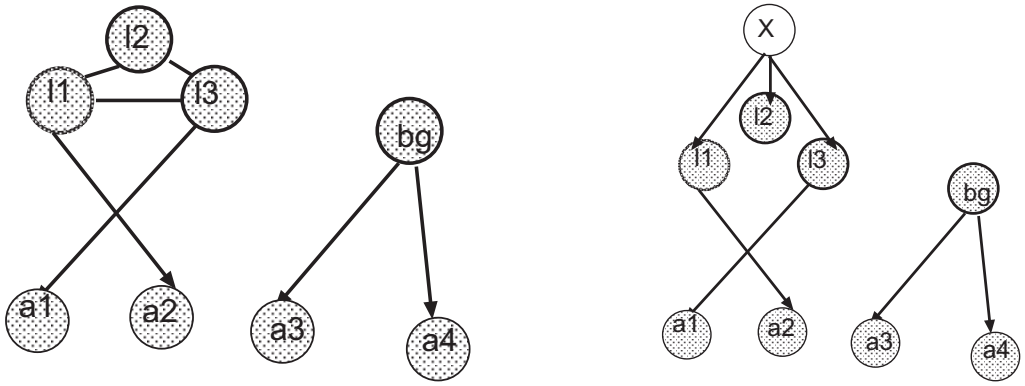

Figure 6: Constellation model as a chain graph in the case that there are $m = 3$ parts, $d = 4$ detections, and the assignment is $\pi_1 = 2, \pi_2 = \emptyset, \pi_3 = 1$. (Left): Fully connected undirected graph between the location nodes, $Y_{1:3}$. Y_1 generates observation a_2, Y_3 generates a_1, and Y_2 is not detected; a_3 and a_4 are generated by the dummy background node. Note that everything is observed (the locations of the parts are assigned to a subset of interest point locations, so $Y_i = l_{\pi_i}$); the model is just used to evaluate the likelihood of this assignment. (Right): A novel latent factor extension.

This exponential complexity has meant that the model has only been used for small numbers of parts ($m \sim 5$) in relatively uncluttered images (with $d \sim 20$). The task has been to classify the image as to whether it contains the object or not (at any location); the model has not been used for localization, because of it cannot easily handle clutter.

Another reason why the constellation model has to have so few parts is the full covariance Gaussian, which has $O(m^2)$ parameters. This can be reduced to $O(m)$ using a latent variable model (see Figure 6(right)). For example, the latent variable X could represent the centroid of the object; this has been called a "common frame" model [MMP04, HL04]. This also makes it easier to use the Hough transform (see below) for detection.

6 Bag of words

One simplification of the constellation model is to throw away the geometry (the joint distribution on $P(Y_{1:m})$). Another more important simplification is to throw away the combinatorial search through assignments, and treat it as a classification problem. Specifically, the task becomes to classify every interest point into one of c classes, and then to somehow use this bag of words representation to solve the object detection task. A popular way to classify interest points is using unsupervised vector quantization. Let $C_i \in \{1, \ldots, c\}$ represent the cluster associated with a_i. This is computed using a nearest neighbor (hard assignment) rule.

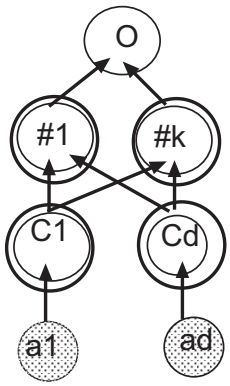

Figure 7: Bag of keypoints as a feedforward circuit. a_i is the the local descriptors at interest point i. $C_i \in \{1, \ldots, c\}$ is the cluster (codebook number) for a_i. d is the number of interest points. $\#k$ is the number of times cluster k occurs. $O \in \{0, 1\}$ indicates presence or absence of the class, *anywhere in the image*.

Due to the influence of the constellation model, and the Caltech databases which they used for evaluation, it has become popular to try and classify the whole image (instead of just a patch) as to whether it contains an object or not, regardless of number or location. This is much easier than having to specify where the object is, since for presence detection, it is sufficient to detect a small, but distinctive piece of the object (eg the wheel of a car).

A popular approach to this image classification task, borrowed from the document classification community, is to compute the codeword counts, and then to apply a classifier (typically an SVM) to the resulting histogram [CDB $^+$04]. This works surprisingly well.

This system can be represented as in Figure 7. Again, this is not really a Bayes net, since C_i is a deterministic function of a_i (nearest neighbor VQ rule), and the histogram is a determinsitic function of the C_i's. Put another way, there is no back-propagation of information, as there would be in a probabilistic model.

7 Implicit shape model

An interesting extension of the bag of words model to the problem of object localization is provided by the implicit shape model [LLS04]. Essentially this memorizes the relative location (wrt the center of the training bounding box) of every feature which is assigned to a given cluster. Then, when the cluster is activated in a test image, it votes for all relative locations (wrt the location where the feature fired in the test image) that are stored in this cluster. This is just like a Hough transform, except each feature can vote for multiple locations: see Figure 8.

Note that, as described so far, this is not a model, but a procedure. However, we may reverse engineer what the underlying model is. In a traditional constellation-type model, one learns a Gaussian star-shaped model of the relative location of parts wrt the center (see Figure 6(right)). The appearance of each part might be a mixture of Gaussians, with each mixture component representing a prototype, but the location is at a fixed offset from the center (modulo some Gaussian fuzz). The ISM turns this on its head, and learns a fixed set of appearances (prototypes), each of which is fixed (modulo some Gaussian fuzz); but the location of each prototype is modeled as a mixture of Gaussians, with each mixture component representing an offset. These two representations are interchangeable: one can define a part as something with a relatively fixed location, but variable appearance, or as something with a relatively fixed appearance, but variable location. Of course, the advantage of defining parts in terms of their location, instead of their appearance, is that one can easily add geometry (i.e., explicit pairwise constraints between parts), which is harder to do in the ISM.

Probably the "right" way to use ISM (from a probabilistic point of view) is as a proposal distribution for importance sampling in a generative model. This way, bottom up suggestions from different detectors can be checked to see if they are globally consistent (eg. they must each explain different pixel data!). Leibe takes steps in this direction by checking that the image at the locations of the Hough transform peaks indeed "look like" the object of interest. He also

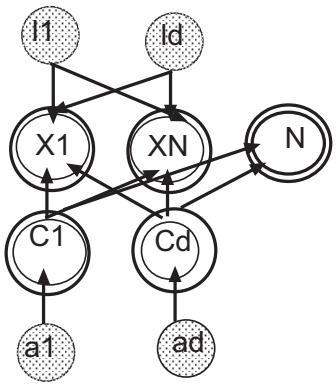

Figure 8: Implicit shape model as a feedforward circuit. d is the number of interest points, N is the estimated number of objects, and $X_{1:N}$ their locations. l_i is the location of the i'th interest point, a_i its appearance, and C_i its nearest cluster center.

uses an MDL criterion to prevent multiple explanations of overlapping pixels. This is clearly a good area for future work.

References

[Bil00] J. Bilmes. Dynamic Bayesian multinets. In *Proc. of the Conf. on Uncertainty in AI*, 2000.

[BSF88] Y. Bar-Shalom and T. Fortmann. *Tracking and data association*. Academic Press, 1988.

[Bun95] W. L. Buntine. Chain graphs for learning. In *Proc. of the Conf. on Uncertainty in AI*, 1995.

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(11), 2001.

[BZ04] A. Barbu and S.C. Zhu. Generalizing swendsen-wang to sampling arbitrary posterior probabilities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2004.

[CDB+04] G. Csurka, C. Dance, C. Bray, L. Fan, and J. Willamowski. Visual categorization with bags of keypoints. In *ECCV workshop on statistical learning in computer vision*, 2004.

[CdFB04] Peter Carbonetto, Nando de Freitas, and Kobus Barnard. A statistical model for general contextual object recognition. In *Proc. European Conf. on Computer Vision*, 2004.

[CS05] Peter Carbonetto and Cordelia Schmidt. Feature selection, data association and spatial integration of cues for object recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2005. Submitted.

[FE73] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. on Computer*, 22(1), 1973.

[FH04] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[FH05] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *Intl. J. Computer Vision*, 61(1), 2005.

[FPZ03] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003.

[GG84] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6), 1984.

[HL04] S. Helmer and D. Lowe. Object class recognition with many local features. In *CVPR workshop on generative models for vision*, 2004.

[HZCP04] Xuming He, Richard Zemel, and Miguel Carreira-Perpinan. Multiscale conditional random fields for image labelling. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[KH03] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[KH04] S. Kumar and M. Hebert. Multiclass discriminative fields for parts-based object detection. In *Snowbird Learning Workshop*, 2004.

[LHB04] Yann LeCun, Fu-Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of CVPR'04*. IEEE Press, 2004.

[LLS04] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proc. European Conf. on Computer Vision*, 2004.

[LMP01] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Intl. Conf. on Machine Learning*, 2001.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. J. Computer Vision*, 60(2):91–110, 2004.

[MMP04] P. Moreels, M. Maire, and P. Perona. Recognition by probabilistic hypothesis construction. In *Proc. European Conf. on Computer Vision*, 2004.

[MPP01] Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):349–361, 2001.

[Pla99] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[PP00] C. Papageorgiou and T. Poggio. A trainable system for object detection. *Intl. J. Computer Vision*, 38(1):15–33, 2000.

[RBK95] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Human face detection in visual scenes. In *Advances in Neural Info. Proc. Systems*, volume 8, 1995.

[TMF04] A. Torralba, K. Murphy, and W. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Info. Proc. Systems*, 2004.

[VJ04] P. Viola and M. Jones. Robust real-time object detection. *Intl. J. Computer Vision*, 57(2):137–154, 2004.