Attention-Driven Parts-Based Object Detection

Ilkka Autio & J.T. Lindgren
Department of Computer Science
University of Helsinki
Finland

Abstract

Recent studies have argued that natural vision systems perform classification by utilizing different mechanisms depending on the visual input. In this paper we present a hybrid, data-driven object detection system that combines parts-based matching and view-based attention for faster detection. We propose a simple competitive policy that allows incremental addition of new object classes to the system without requiring class-vs-class training. Using our framework, we show empirical support for the hypothesis that low-frequency visual information can be effectively used to direct attention and possibly subsume further, more costly analysis. We evaluate our approach on face and car detection problems, while concentrating on the capability to learn from small samples. Our implementation is freely available as Matlab source code.

1 Introduction

Despite significant amount of research directed at devising analytically good algorithms for various low-level tasks on images, it remains unclear how the methods should be used to allow machines to succeed in high-level tasks such as object detection and image understanding. Meanwhile, the vision research community has been converging towards emphasizing integration [9]. One major paradigm of vision is that veridical perception arises from data- and expectation-driven interaction of the various parts of the system. The individual pieces of the neural machinery may be operating reasonably only statistically [9] or converge to some solution relying on feedback from the other levels in the visual system [7, 2].

In this paper we work towards integrating different types of methods by proposing a system for object detection that combines slow parts-based matching with a fast attention mechanism[1]. In our system, the attention mechanism is used to quickly select in a data-driven fashion and per-view basis what more precise further mechanisms need to be activated, if any. We will empirically show how this mechanism is able to enhance detection and training speed significantly without adverse effects in accuracy. Also, we will propose a simple competitive policy to extend our system to multi-object

[1] To ensure replicable research, the source code for our implementation is available at `http://www.cs.helsinki.fi/group/porr/`

detection without need for class-vs-class training, allowing new object categories to be added to the system incrementally. It should be noted that whereas the modular form and the policy of our system are fixed, the individual components are still estimated from the data, i.e. learned statistically from labeled training examples.

Our system allows us to look for empirical support for an interesting open problem in vision research. Recent work by Bar [2] examines the question of how biological vision systems can classify familiar scenes and objects very fast. Bar argues that it is possible for a higher-level visual component to receive a low-frequency (LF) representation of the scene (i.e. a blurred image) to use in fast decision making. In our experiments we will show that in our system it is possible to train and evaluate the attention mechanism on low-frequency images, without notable reduction in accuracy of the whole system. This gives positive support for the LF hypothesis.

Another interesting issue is the ability to learn to recognize objects from only a few examples. From the viewpoint of learning theory, this may be caused by suitable "bias" present in natural learners. Previous work [11, 12] suggests that the fragment features used by our parts-based component are well-biased towards visual detection tasks. We will show empirical support for this hypothesis by comparing our method to some state-of-the-art black-box learning methods.

To put our work in historical context, recent vision research supports the idea that natural perception is at least partly data-driven [9, 2, 7, 10]. Papers by e.g. Bar [2] and Torralba and Oliva [10] show support for the idea that low-frequency or coarse statistical information could in some cases suffice for decision making. Lee and Mumford [7] go even further and argue for hierarchical inference in the cortex, where components of the visual system would be communicating with probabilistic information until convergence to some likely interpretation of the scene.

On the side of object detection, a few systems have been recently proposed that use some type of data-dependent rejection mechanism [13, 5]. For example, the system of Viola and Jones [13] learns a rejector cascade of simple wavelet-like features that can often skip negative (non-class) image patches quickly.

Our system differs from the previous work in that it retains the good aspects of the parts-based approach as in [11, 12] while being faster and applicable in a multi-object setting. Like [13], our system classifies easy examples faster than difficult ones. Unlike the cascade method, our features range from simple to complex. Also, our empirical finding that attention could potentially be achieved with a very simple mechanism is of separate interest.

The rest of this paper is organized as follows. In section 2 we describe our hybrid method in detail and show justification for the used prediction policies. Section 3 relates our experimental design and describes the used datasets. We present our empirical results in section 4. Finally, section 5 concludes with some future directions.

2 The hybrid method

The hybrid method presented in the current paper combines the advantages of quick view-based classification with a more accurate parts-based matching. We will describe the parts-based module first.

2.1 Robust detection module (RDM)

A parts-based mechanism is used to classify the most challenging images, i.e. those that require the most attention. It is a modification of the scheme first proposed in [11, 12]. There, the mechanism is primarily intended to demonstrate, as a proof of concept, that intermediate complexity features are better suited to visual discrimination than low-level features such as simple wavelets.

The learning algorithm extracts $I \in \mathbb{N}$ highly specific bitmap templates (fragments) of object parts from the in-class training data $\mathcal{T}_p$. The modeling assumption is that characteristic spatial arrangements of these parts determine the object class. The model is best suited for semi-rigid objects (e.g., faces) having fairly rigid parts, (e.g., eyes).

In practice, each fragment is associated with the rough image location it was extracted from. The fragments are matched using normalized cross-correlation. The degree of a match is compared to a fragment-specific threshold: If the maximum match value around the rough location exceeds the threshold, the fragment is present.

Fragments are selected for informativeness, as measured by class conditional entropy. The first fragment $F^{(1)}$ is selected to maximize

$$H(C) - H(C|F), \tag{1}$$

where $H(C)$ denotes the Shannon entropy of the class, and $H(C|F)$ is the class entropy given the status of the fragment F is known (i.e., present or absent). Subsequently, we select the fragment $F^{(t)}$ maximizing the additional information

$$min_{j<t}(H(C|F^{(j)}) - H(C|F^{(j)}, F^{(t)})). \tag{2}$$

The greedy selection algorithm is

1. $S :=$ image locations for promising candidate fragments in $\mathcal{T}_p$.
2. $Fset :=$ candidate fragments sampled around locations in S.
3. For each $F_i \in Fset$, choose the matching threshold that maximizes equation 1.
4. Build a binary occurrence table O, in which $O(i, j) = 1$ iff the i:th candidate fragment F_i occurs in the j:th image. Equations 1 and 2 can be evaluated using O.
5. Allocate a gain table G. The table will be incrementally filled in the next loop with $G(i, j) = H(C|F_i) - H(C|F_j, F_i)$.
6. For $t := 1$ to I, do
   Choose $F^{(t)}$ according to equations 1 and 2. Fill in G as new gain estimates become available, and use G to avoid re-evaluations of equation 2 for known pairs.

Comparing the above to [11, 12], we see some differences: First, we have replaced the brute-force search with an attention mechanism that provides a set S of good sampling locations in $\mathcal{T}_p$ (elaborated in the next subsection). Second, we introduced a cache

G for speeding up the fragment selection loop. The attention mechanism lets us concentrate on far fewer candidate fragments than would otherwise be possible. Hence, a simple static table G becomes a feasible cache. Otherwise, it would be best to cache just the column minima of G: in terms of equation 2, we would then keep track of the "worst opponent" of each candidate, making the algorithm a bit more complex.

It is easy to see that caching (in any sane form) is useful, as evaluating equation 2 is expensive, having a computational cost proportional to the size of the training data. In the following, we will briefly analyze the effect of caching to the parts-based algorithm.

Let $N = |Fset|$ be the initial number of candidate features and I be the number of features we desire ($N >> I$). After the first iteration, we must use formula (2) and test each of the remaining $(N - (t-1))$ candidates against each of the $t-1$ previously selected ones. The overall number of pairwise evaluations is

$$k_1 \approx N\frac{I-1}{2}I - \int_1^{I-1} t^2 dt \tag{3}$$

Using a reasonable caching scheme, such as our G, no pair has to be evaluated twice, and the number of such evaluations becomes

$$k_2 = N(I-1) - \frac{I-1}{2}I. \tag{4}$$

When the values of N and I are reasonable, it is easy to see that typically $k_1 >> k_2$.

After selecting I fragments, we use the fragments as binary features (i.e., present or not) shown to a linear discriminator. The discriminator then classifies the images and is responsible for the output of a *robust detection module* (RDM). The discriminator is learned using the same method as the attention controller that we describe in the next subsection. The attention module provides the sampling locations for RDM learning, and controls the activation of the RDM.

2.2 Attention controller module (ACM)

An *attention controller module* (ACM) serves two purposes: it assists in training the corresponding RDM by providing the required sampling locations, and controls the selective activation of the RDM.

Some intuitive requirements can be given for an attention mechanism: It should be able to make fast preliminary predictions, and give rates of confidence for the predictions. A reliable confidence estimation is required for controlling the RDM. It might be of additional interest if the attention model could be formulated in a neurally plausible way. In the following, we will propose a simple linear attention model to meet these requirements.

We use a global, view-based approach. Given a similarity measure $s(\mathbf{x}_i, \mathbf{x}_j) \in \mathbb{R}$ between two views $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$, where $\mathbf{x}_i \in \mathbb{R}^k$ is a data vector and $y_i \in \{-1, 1\}$ its binary affiliation w.r.t. some target class c (e.g., "cars"), we can set up a class-membership function $f_c(x) : \mathbb{R}^d \to \{-1, 1\}$ as

$$f_c(\mathbf{x}) = sign(\sum_i y_i s(\mathbf{x}_i, \mathbf{x})), \tag{5}$$

Figure 1: A side view car image and the highlighted regions around the pixel locations in S, as given by the most significant coefficients of $\mathbf{w}$.

Now $f_c(\mathbf{x})$ has a view-based form, but is slow to evaluate. However, it can be shown (see e.g. [4]) that the expression is equivalent to a dual form linear classifier,

$$f(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} + b) = sign(\sum_i y_i \alpha_i \mathbf{x}_i^T \mathbf{x} + b), \tag{6}$$

if a simple dot product is used as the similarity measure $s(.,.)$ between the examples. In equation 6, $b \in \mathbb{R}$ is an estimated parameter of the model and the $\alpha_i \in \mathbb{R}^+$ reflect that not all views are necessarily weighted equally. The important point is that a single weight vector $\mathbf{w} \in \mathbb{R}^d$ incorporates information from possibly all the given example views $\mathbf{x}_i$ and is still in essence view-based.

An established way to learn such a hyperplane from labeled data is to use the Support Vector Machine (SVM) framework [4]. We now briefly describe the SVM variant we used. With the previously defined notation and l as the size of the training set $\mathcal{T}$, the optimization criteria for a 1-norm *soft margin* linear SVM can be specified as

$$\begin{array}{ll} \text{minimize}_{\xi,\mathbf{w},b} & \mathbf{w}^T\mathbf{w} + \mathcal{C}\sum_{i=1}^{l}\xi_i, \\ \text{subject to} & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \forall i, \\ & \xi_i \geq 0, \forall i, \end{array}$$

where $\xi \in \mathbb{R}$ are the slack variables related to misclassifying the respective example, and $\mathcal{C} \in \mathbb{R}^+$ a misclassification cost parameter given as input. The optimization attempts to find a regularized hyperplane $(\mathbf{w}, b)$ separating the classes while allowing for occasional classification errors, as controlled by $\mathcal{C}$. For a more detailed description, see e.g. [4].

Empirically, it turns out that the estimated linear classifier $(\mathbf{w}, b)$ can be used for an ACM. First, it can work as an attention mechanism in the learning phase, and the sampling locations for the RDM can be found just by looking at the most significant coefficients of $\mathbf{w}$. This is illustrated in figure 1. Second, the distance of each example from the separating hyperplane can be calculated efficiently and used as a confidence value. If the ACM is confident, the RDM does not have to be activated. Evidence for this behavior can be seen in figure 2, which shows the distribution of case distances from a learned hyperplane in a car detection problem. Altough the linear classifier cannot separate the classes perfectly, the distributions are suitable for the use of simple thresholding to determine regions where the classifier is accurate, and where it will abstain and defer to the RDM. The thresholds could be estimated from a separate tuning set, but here we used the medians of the distances of the examples in the training set.

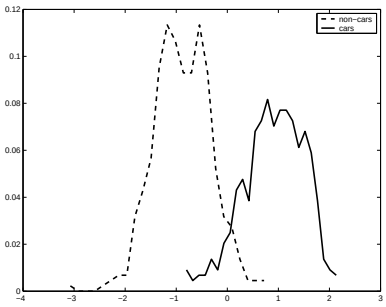

Figure 2: Distribution of distances from the discriminating view-based hyperplane $(\mathbf{w}, b)$ on the car detection problem.

2.3 Detection in a single-object setting

The ACM and RDM modules should be combined in a disciplined way that allows high accuracy and speed. We will first propose a detection policy for a single object case:

1. The object-specific ACM sees an image and makes a prediction.
2. If the ACM is not confident, it abstains and activates the corresponding RDM, which chooses the final prediction. Else, the prediction of the ACM is final.

The analysis is straightforward. Let $d(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ stand for the raw ACM output. We abbreviate this by d, because $\mathbf{x}$ is not used directly in the following. Similarly, let D stand for the raw RDM output (i.e., it is also a linear machine having similar functional form). We assume that $P(D|d, y) = P(D|y)$, i.e., the RDM output is independent of the ACM output *if* the class is known. Without loss of generality, we assume for notational simplicity that the confidence regions of the ACM can be represented by a single value $T \in \mathbb{R}^+$, i.e., the ACM is confident iff $|d| > T$. For brevity, we abbreviate conditional probabilities $P(A|B)$ as $P_B(A)$, and the probability of correctness as $P(ok)$. Note that $P(ok|D, |d| < T) = P(ok|D)$, because when D becomes known, the system makes the prediction based on that value alone. The probability of a correct prediction is

$$P(ok) = P(|d| > T)P_{|d|>T}(ok) + P(|d| < T)P_D(ok). \tag{7}$$

We can view equation 7 as a function of T:

$$acc(T) = \beta(T)\gamma(T) + (1 - \beta(T))\delta. \tag{8}$$

When T grows, the event $|d| > T$ becomes less likely and $\beta(T) = P(|d| > T)$ diminishes. Also, the nature of the data and the optimization criteria allows us to assume that the classification of examples having $|d| > T$ becomes more accurate as T grows, increasing $\gamma(T) = P_{|d|>T}(ok)$ as well (see figure 2). We observe that

- If $\exists T$ s.t. $\gamma(T) > \delta > \gamma(0)$, and $\beta(T) > 0$, then the hybrid accuracy exceeds both the RDM and ACM. The ACM becomes a *specialist* of a subset of examples.
- With a smaller T, we may have $\gamma(T) = \delta$, and a larger $\beta(T)$, thus gaining speed without losing accuracy. In practice $\beta(T) > 0.4$.
- We can sacrifice accuracy for speed by allowing a very low T s.t. $\gamma(0) < \gamma(T) < \delta$.

Next, we extend the policy to detect multiple objects.

2.4 Extension to a multi-object setting

In a multi-object setting we have m object classes and a single *no object* class. Each of the m object classes has an ACM and a RDM trained to discriminate the object from the *no object* class.

For our multi-object policy to work, the classes should be distinct enough to allow coarse discrimination on the basis of low-frequency information. However, many conceivable classes differ also in their details. A typical RDM searches for highly specific object parts, such as car tires, which seldom appear in other classes. Thus, this property of the domain allows us to potentially skip class-vs-class -training.

Each of the m ACMs must select an answer from the set $\{yes, abstain, no\}$. Each activated RDM selects from $\{yes, no\}$. The mutually exclusive detection policy is as follows,

1. If all m ACMs say no, then $prediction =$*no object*.
2. Else, if a single ACM$_j$ says yes, then $prediction = j$.
3. Else, if more than one ACM says yes, the predicted class is selected randomly from the positive matches (practically rare).
4. Else, let A be the set of object classes whose ACMs abstained. For each class $i \in A$, activate the corresponding RDM and determine the predicted class: If all say no, then *no object*. Else, if more than one yes, randomize among the positive matches (practically rare). Else, the single yes determines the prediction.

To allow analysis, we must elaborate. We denote the object classes as $1, \ldots, m$, and *no object* as -1. Class priors are $P(i) > 0$ for all classes i. The event $d_i > T_i$ means ACM$_i$ predicts yes, and $d_i < -T_i$ means no. Else, ACM$_i$ $abstains$. The events $D_i > 0$ and $D_i < 0$ for the RDMs are interpreted the same way. Classifier outputs are assumed independent if the class is known. To simplify the analysis, we pessimistically assume that the randomized predictions are always wrong. An image is classified correctly iff one of the following conditions is true,

- An object i is detected *fast*, if $d_i > T_i$ and $\forall j \neq i \Rightarrow d_j < T_j$.
- An object is detected *slowly*, if first $|d_i| < T_i$ and then $D_i > 0$, *and* $\forall j \neq i \Rightarrow d_j < T_j$ *and* $\forall j \neq i, RDM_j$ active $\Rightarrow D_j < 0$.

- A non-object is dismissed, i.e., no module says yes.

Abbreviating $P_i(D_i > 0)$ as $P_i(D_i)$, we get

$$\begin{aligned}
P(ok) = &\quad \textstyle\sum_{i>0} P(i)P_i(ok) + P(-1)P_{-1}(ok) && (9)\\
P_i(ok) = &\quad P_i(d_i > T_i)R_i + P_i(|d_i| < T_i)P_i(D_i)Dis_i && (10)\\
P_{-1}(ok) = &\quad \textstyle\prod_{j>0} P_{-1}(dis_j) && (11)\\
R_i = &\quad \textstyle\prod_{j>0:j\neq i} P_i(d_j < T_j) && (12)\\
Dis_i = &\quad \textstyle\prod_{j>0:j\neq i} P_i(dis_j) && (13)\\
P_i(dis_j) = &\quad 1 - P_i(d_j > T_j) - P_i(|d_j| < T_j)P_i(D_j), && (14)
\end{aligned}$$

where the (dismissal) dis_j means there is no false detection of class j. Next, we prove that the policy may, in principle, exceed the accuracy of a committee of RDMs.

We make some reasonable assumptions about the thresholds. Well-behaved ACMs have $P_i(d_j > T_j) << P_i(d_j < -T_j)$, and $P_i(d_j < -T_j) > 0$. In addition, $P_i(d_j > T_j) < P_i(D_j > 0)$, i.e., increasing T_j can make early false detections rare. If the accuracies of the RDMs are limited by (let $\epsilon > 0$,$j \neq i$)

$$\begin{aligned}
P_i(D_i) &< \frac{P_i(d_i > T_i)}{P_i(d_i < -T_i) + P_i(d_i > T_i)} && (15)\\
P_i(D_j) &> \frac{P_i(d_j > T_j)}{P_i(d_j < -T_j) - \epsilon}, && (16)
\end{aligned}$$

then the hybrid exceeds the accuracy of the committee of RDMs. To see this, we first take the trivial inequality (again $i \neq j$)

$$P_i(D_j) > \quad \big[P_i(|d_j| < T_j) + P_i(d_j < -T_j) - \epsilon\big]P_i(D_j). \tag{17}$$

Multiplying (17) by -1, adding 1 to both sides, and finally substituting bound (16) yields

$$P_i(dis_j) > P_i(D_j < 0). \tag{18}$$

It immediately follows from (18) that

$$Dis_i = \prod_{j>0:j\neq i} P_i(dis_j) > \prod_{j>0:j\neq i} P_i(D_j < 0) = Dis_i^*. \tag{19}$$

The new term Dis_i^* is the probability that the individual machines of a pure RDM committee do not make false detections.

Next, performing simple algebraic manipulation of the assumption $P_i(d_j > T_j) < P_i(D_j > 0)$ yields

$$P_i(d_j < T_j) > P_i(D_j < 0). \tag{20}$$

Using (20), we immediately see that

$$R_i = \prod_{j>0:j\neq i} P_i(d_j < T_j) > \prod_{j>0:j\neq i} P_i(D_j < 0) = Dis_i^*. \tag{21}$$

Substituting the inequalities (19,21) into (10) yields

$$\begin{aligned} P_i(ok) > & \quad \left[P_i(d_i > T_i) + P_i(|d_i| < T_i) P_i(D_i)\right] Dis_i^* && (22)\\ > & \quad \left[P_i(|d_i| > T_i) + P_i(|d_i| < T_i)\right] P_i(D_i) Dis_i^* && (23)\\ = & \quad P_i(D_i) Dis_i^* && (24)\\ = & \quad P_i^*(ok), && (25) \end{aligned}$$

where (23) resulted from substituting the bound (15) into (22). The new term $P_i^*(ok)$ is the probability that a pure RDM committee predicts correctly if the true class is $i > 0$. Finally, substituting (18) into (11) gives the corresponding result for the *no object* class

$$P_{-1}(ok) > \quad P_{-1}^*(ok). \tag{26}$$

Substituting (25) and (26) into (9) shows that if the bounds (15) and (16) hold, the hybrid is more accurate than a pure RDM committee. If the bounds are relaxed, we can trade accuracy for speed.

3 Experimental design

We evaluated our method on two different problems: car and face detection. The car data from Agarwal and Roth [1] has about 1000 gray-scale images in 100×40 resolution. Our face dataset is more complicated, as we randomly sampled the face images from the AR dataset [8] which we then embedded on larger backgrounds sampled from the BioID database [6], which we also used for the non-face images. This was to make the problem more challenging for our hybrid method. The modified face dataset has about 1900 gray-scale images in 200×40 resolution. Both datasets were balanced to contain approximately equal amount of in-class and out-class images.

To evaluate the low-frequency hypothesis, we performed two experiments, where we convolved the images with Gaussian filters. First, we used a 7×7 mask with standard deviation $\sigma = 1$. In the second experiment, we applied a 9×9 mask with $\sigma = 2$ for heavier blurring. The blurring approximates discarding high-frequency information. Unlike the ACM, the RDM was allowed to use the original data, as is acceptable from the viewpoint of the LF hypothesis.

In the following, we use some abbreviations. A1 is our hybrid approach where the ACM is used only for the selection of fragments in the learning phase. A2 is the same, except that the ACM is also used in the evaluation phase for fast classification of easy instances. RP is a pure RDM that selects fragments from a random set of candidates sampled from the in-class data. When RP is compared to the hybrid, the random set size equals the size of the $Fset$ used by the hybrid. SVM1 and SVM2 are linear and second-order polynomial soft margin Support Vector Machines, respectively. For both, we used cost-constraint $\mathcal{C} = 1$. RF80 denotes Random Forests [3] using 80 full-grown decision trees. These three methods are state-of-the-art machine learning algorithms and operate here on raw image data.

The settings of the components of the hybrid method were as follows. The candidate fragment size in all RDMs was fixed to 16×16 pixels and the number of fragments

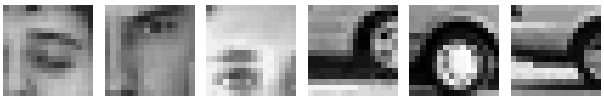

Figure 3: Some example fragments A2 extracted from faces and cars

I was set 50. For the ACMs, we chose the locations of the 60 most significant coefficients (out of over 4000) of $\mathbf{w}$ as the seed locations. The RDM subsequently sampled 5 fragment candidates from each location (i.e., it chose 5 positive images randomly, and from each, extracted a fragment centered on the seed pixel). We chose the thresholds of the confidence regions by simply selecting the medians of $\{d(\mathbf{x}_i) : y_i = 1\}$ and $\{d(\mathbf{x}_i) : y_i = -1\}$ calculated from the training set.

The results reported describe percentage-correct accuracies on *inverted* 10-fold cross-validation. Instead of using $(n-1)/n$ of the data for training and $1/n$ for testing as per iteration in normal n-fold cross-validation, we inverted the roles of the training and testing sets to measure how the methods could learn from small training samples.

For testing the hybrid method in a multi-object situation, we cropped the face data to the same resolution as the car data. Then, for each fold of the inverted cross-validation, we trained both car and face detectors with their respective, disjoint training folds. To create the respective test fold, we combined the disjoint test folds of that iteration into one three-class fold. The combined set was then used to evaluate the A2, now using the multi-object policy of subsection 2.4. The multi-class results of the other methods were obtained using multiple binary classifiers and a *one-against-all* wrapper.

4 Results

Our results are shown in table 1. In the car problem, A1 using the attentive mechanism has similar accuracy to RP, the basic parts-based method. In the face problem, A1 seems to have an advantage over RP. Here, the actual face area is small compared to the background area, while the car images contain relatively little background. The precise framing of the selected parts does not matter much (see figure 3). Attentive selection seems useful when there is more to select from, i.e., when the important parts do not dominate the images. At least, there are no apparent disadvantages.

Comparing A2 to A1 shows that utilizing the simple linear machine in the evaluation phase did not degrade the accuracies. In the car problem, the ACM of A2 was about 770 times faster than the RDM, and in the face problem, about 340 times faster. We denote the constant evaluation times of the modules by t_{ACM} and t_{RDM}. The ACM is confident with probability p, and the expected evaluation time of A2 in the single-object setting is:

$$E[t_{A2}] \quad = \quad pt_{ACM} + (1-p)(t_{ACM} + t_{RDM}). \tag{27}$$

In the car problem p was about 0.44 and in the face problem it was about 0.42. Substituting observed values into (27) and calculating $E[t_{A2}]/t_{RDM}$ gave us the relative

Table 1: Inverted 10-fold cross-validation accuracies.

	A1	A2	RP	SVM1	SVM2	RF80
Cars	0.9681	0.9702	0.9506	0.9263	0.9378	0.9247
Faces	0.9679	0.9691	0.9303	0.9019	0.8787	0.9224
Multi	-	0.9483	-	0.9030	0.9009	0.9190
$C_{\sigma=1}$	-	0.9664	-	0.9194	0.9349	0.9239
$C_{\sigma=2}$	-	0.9605	-	0.9090	0.9272	0.9167
$F_{\sigma=1}$	-	0.9616	-	0.8912	0.8710	0.9161
$F_{\sigma=2}$	-	0.9545	-	0.8738	0.8568	0.9066

values of 0.56 and 0.58 for the two problems. On the average, classifying a car using A2 costs just 56% of the pure RDM cost, because 44% of the cars are classified instantly using ACM only. While the precise values are implementation specific, a reasonable advantage in speed is maintained if the ACM is at least an order of magnitude faster than the RDM. Recalling that the ACM computes a simple dot product of an image, while the RDM must compute normalized cross-correlation of dozens of fragment templates, the speed advantage seems useful in practice.

Dataset "multi" in table 1 denotes the multi-class experiment with faces and cars. Here A2 is again somewhat better than the black-box methods. As for the run-time behaviour of A2, for approximately $1/3$ of the test cases, RDMs weren't used at all, while for another $1/3$ one of the RDMs was required, and for the remaining $1/3$, both RDMs were executed. The usage frequencies of the RDMs were nearly symmetric for both object classes. A calculation similar to (27) shows that in this problem A2 is expected to take about 50% of the time required by a pure RDM commitee.

Finally, the results of the LF experiment are shown in the lower half of table 1. It can be seen that blurring the data caused only graceful degradation for all of the methods. This supports the LF hypothesis, and that to some extent general visual categories might be recognizable with LF information alone. Also, if the ACM is used in lower resolution, the hybrid speed advantage is emphasized.

5 Conclusion

We proposed a hybrid, multi-class object detection system combining parts-based matching with fast attention, and showed empirical support for the hypothesis that low-frequency visual data could be used for attention.

It would be interesting to evaluate our method and policy on datasets with more different object classes, possibly revealing practical weaknesses in our policy, such as scalability issues. Perhaps a more dynamic policy could be estimated from data. Also, the expressive power of the current system is somewhat limited. It is clear that the RDM is unsuitable for classifying textures or deformable shapes that have no rigid parts. In addition, it has no explicit internal mechanisms for handling rotation or scaling. ACM, on the other hand, requires centered training data. A possible future direction would be to use the RDM module to select, label and center the data while training the ACM

module. Finally, both RDM and ACM are batch methods that might be replaceable by on-line algorithms.

References

[1] S. Agarwal and D. Roth, 'Learning a sparse representation for object detection', in *Proc. of the 7th European Conference on Computer Vision*, (2002).

[2] M. Bar, 'A cortical mechanism for triggering top-down facilitation in visual object recognition', *Journal of Cognitive Neuroscience*, **15**, 600–609, (2003).

[3] L. Breiman, 'Random forests', *Machine Learning*, **45**(1), 5–32, (2001).

[4] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.

[5] M. Elad, Y. Hel-Or, and R. Keshet, 'Rejection based classifier for face detection', *Pattern Recognition Letters*, **23**(12), 1459–1471, (2002).

[6] O. Jesorsky, K. Kirchberg, and R. Frischholz, 'Robust face detection using the Hausdorff distance', in *Proc. of the 3rd International Conference on Audio and Video based Person Authentication*, pp. 90–95. Springer, LNCS-2091, (2001).

[7] T. S. Lee and D. Mumford, 'Hierarchical bayesian inference in the visual cortex', *Journal of the Optical Society of America*, **20**(7), 1434–1448, (2003).

[8] A.M. Martinez and R. Benavente, 'The AR face database', Technical report, CVC #24, (1998).

[9] S.E. Palmer, *Vision Science*, The MIT Press, 1999.

[10] A. Torralba and A. Oliva, 'Statistics of natural image categories', *Network: Computation in Neural Systems*, **14**, 391–412, (2003).

[11] S. Ullman, M. Vidal-Naquet, and E. Sali, 'Visual features of intermediate complexity and their use in classification', *Nature Neuroscience*, **5**(7), 682–687, (2002).

[12] M. Vidal-Naquet and S. Ullman, 'Object recognition with informative features and linear classication', in *Proc. of the 9th IEEE International Conference on Computer Vision*, (2003).

[13] P. Viola and M. Jones, 'Rapid object detection using a boosted cascade of simple features', in *Proc. of the 2001 IEEE Computer Society Conference on Computer vision and Pattern Recognition*, pp. 511–518, (2001).