Efficient Object Detection via Adaptive Online Selection of Sensor-Array Elements

Matthai Philipose
Microsoft

Abstract

We examine how to use emerging far-infrared imager ensembles to detect certain objects of interest (e.g., faces, hands, people and animals) in synchronized RGB video streams at very low power. We formulate the problem as one of selecting subsets of sensing elements (among many thousand possibilities) from the ensembles for tests. The subset selection problem is naturally *adaptive* and *online*: testing certain elements early can obviate the need for testing many others later, and selection policies must be updated at inference time. We pose the ensemble sensor selection problem as a structured extension of test-cost-sensitive classification, propose a principled suite of techniques to exploit ensemble structure to speed up processing and show how to reestimate policies fast. We estimate reductions in power consumption of roughly 50x relative to even highly optimized implementations of face detection, a canonical object-detection problem. We also illustrate the benefits of adaptivity and online estimation.

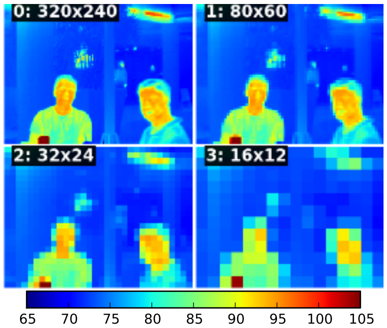

Figure 1: Output from a 4-level FIR Ensemble. A *single* temperature threshold check at a small number of pixels (e.g., image 3 above has 400x fewer pixels than image 0!) suffices to reject most pixels. When possibly interesting pixels appear, finer-grained sensors may be used *adaptively* to confirm the object. E.g., image 2 may be used to find the whole head and image 1 to then confirm eye and nose temperatures.

Consider face detection on video streamed from a wearable device. The standard detection algorithm, due to Viola and Jones (Viola and Jones 2004), computes local features for every window in every video frame at various scales (while taking care to stay efficient by avoiding recomputing features) and classifies every window using a cascaded binary classifier that on average performs a dozen multiplications and additions on each window. Algorithms for detecting hands, objects and pedestrians have a similar *windowed feature matching* structure (Dalal, Triggs, and Schmid 2006). Proposed silicon implementations of the Viola-Jones algorithm would consume 600-800mW (Hori and Kuroda 2007; Aptina 2011) to process 10 frames/sec of 180° field-of-view (FOV) video. Given a realistic budget of roughly 7mW (based on a generous fraction of a 200mAh battery), we seek efficiency improvements approaching 100x *even relative to silicon implementations*.

Our solution rests on two observations. First, as Viola and Jones originally noted, most pixel windows do not contain objects of interest (e.g., over 99% of windows in our day-to-day first-person video dataset contain no faces). Second, given *gating* imagers that measure quantities (e.g., temperature or depth) other than light intensity over the same field of view as the video imager, a single (low-resolution and therefore low-power) scalar measurement in each window could establish the absence of the object. For instance, if a sensor reported average temperature over a window, a single threshold check for skin temperature range (30-35°C) may suffice to reject the window. The scalar threshold check avoids the standard per-window featurization and classification overhead. If every window in a frame fails the threshold check, it is unnecessary to even access the RGB imager, substituting for it the lower cost of accessing the lower-resolution gating imager. In fact, even when some windows contains target objects, given recent "proportional power" (LiKamWa et al. 2013) imagers, it may be possible to pay just for accessing these windows in the frame.

As our gating sensor, we consider hierarchical ensembles of thermographic (or "far-infrared (FIR)") imagers (Figure 1). These *FIR imagers* (Dereniak and Boreman 1996) cover the field of view of the traditional *light imager* being gated at progressively finer resolution. They report average temperature over the FOV of each of their pixels, instead of illumination intensity as in video. Since gating imagers are typically lower resolution than the light imager, each pixel in the former corresponds to many windows in the latter. Given the stringent power budget, it is necessary to select a small subset of the several thousand *gating pixels* for checking. Reading certain pixels can remove the need to read others; e.g., a coarse pixel that returns a low average temperature may preclude a face in its FOV.

We formulate the problem of selecting such sets of gating pixels as that of estimating and evaluating a "shadow" classifier, in an adaptive (i.e., gating pixels are selected sequentially, with early choices influencing later ones) and online (i.e., selection policy is re-estimated as frames are received) setting. We extend work over the past two decades on cost-sensitive classification to the case of large arrays of sensors and their online re-estimation. To keep costs of processing the gating imagers low, we describe a suite of optimizations that exploit ensemble structure. We present empirical evidence over several hundred thousand frames of temperature-gated video from a variety of day-to-day settings that shows an estimated reduction of 50x in power required to detect faces relative to RGB-only processing, at 9% reduction in detection rates. We further break out the benefits of adaptivity, online processing and our optimizations.

Related Work

The computer vision community has considered using FIR imagers as additional features for improved pedestrian (Zhang, Wu, and Nevatia 2007) or face recognition (Wolff, Socolinsky, and Eveland 2005; Socolinsky and Selinger 2004), or as the basis for biometric identification (Buddharaju et al. 2007). We shift the focus to improved detection efficiency, using FIR imagers as an early stage in the detection pipeline. Further, we consider emerging (LiKamWa et al. 2013) *proportional power* implementations of imagers, where accessing only parts of images is rewarded by lower power consumption. These goals motivate our focus on sequential sensor-selection as opposed to better classification or physiological modeling.

The canonical approach to detecting objects efficiently is that of Viola and Jones (Viola and Jones 2004). They apply highly efficient degenerate decision trees over every window in a grayscale image, showing how to classify millions of variables in real time. More generally, but in a small-variable-set setting, reducing the cost of classification is the goal of *cost-sensitive classification* (Yang et al. 2006; Turney 1995; Zubek and Dietterich 2002; Greiner, Grove, and Roth 2002), often motivated by selecting inexpensive but effective medical diagnosis steps. In this setting, subcomputations (e.g., sensing, feature extraction, classification) are assigned a cost, and the overall classifier is structured to optimize loss functions (often based on *value of information* (Howard 1966) or misclassification rates) under a budget. In some cases, the output of training is an explicit decision tree (Turney 1995; Benbasat and Paradiso 2007; Xu et al. 2013) where comparisons that provide high value of information appear higher in the tree. In others (Gao and Koller 2011; Yang et al. 2006; Zheng, Rish, and Beygelzimer 2005), training yields a joint distribution of sensor values and classification results that, at test time, is greedily used to select sensors sequentially based on expected increase in objective conditioned on sensors selected thus far. To the best of our knowledge, none of this work has considered the extreme efficiency required to handle large structured arrays of gating sensors, or online re-estimation. We propose a suite of optimizations to the underlying value-of-information-based approach to make this setting tractable.

The theory of optimizing *monotone submodular* functions in adaptive (Golovin and Krause 2011) and online settings (Streeter and Golovin 2008) provides formal guarantees for problems very similar to adaptive sensor selection and inspired our work. However, the commonly used objective function for adaptive sensor selection is not even monotone. Several further assumptions used in the adaptive online setting (Gabillon et al. 2013) such as independent sensors, objective function independent of the joint probability distribution of sensors, are impractical when applied to sensor-selection. We hope to motivate stronger results in this area.

Setup: Ensemble Shadow Classifier Learning

Let $f_1, \ldots, f_T$, $f_t \in F = \mathbb{R}^{m \times n}$ be a sequence of *frames* observable by a *primary imager*. With each frame f_t, we associate a set $W^t = \{W^t_{ijs}\}$ of variables called *windows*. Let $\mathcal{W} = \cup_t W^t$ and n_s be the number of windows of size s in each frame. Window W^t_{ijs} of *size* $s \in S \subseteq \mathbb{N}$ *covers* elements $f_{t[i:i+s][j:j+s]}$ of f_t; we write $f[w]$ for the elements of f covered by window w.

Let $h_p : \mathcal{W} \times F \rightarrow \{0, 1\}$ be a classifier (which we call the *primary classifier*) that determines, for each window, whether its covered elements constitute an object of interest. Let $c_r \in \mathbb{R}$ be the *cost* of reading a single element of a frame, c_c be the cost of classifying a window[1], and $B^* \in \mathbb{R}$ the *budget* for processing all frames (a budget of $B = \frac{B^*}{T}$ per frame in expectation). Typically, elements are read just once but classified multiple times as part of different windows. Classifying every window in a frame costs $C_p = mnc_r + |W^t|c_c$. As sketched in the introduction, this cost may far exceed the budget: $C_p > 100B$ is plausible.

In order to reduce power consumption, we seek to use the output from an *ensemble of gating imagers*, $\Gamma = \gamma_1, \ldots, \gamma_M$ ($\gamma_i \in \mathbb{R}^{m_i \times n_i}$) that have the same field of view as the primary imager. Imagers in the ensemble have progressively lower resolution: $m_i, n_i < m_j, n_j$ for $i > j$. Typically, gating imagers have much lower resolution than the primary: $m \gg m_i$. We write g_{ijmt} to denote the (i, j)'th element of γ_m at time t, with $g_t = \cup_{i,j,m} g_{ijmt} \in G$. We omit index t sometimes for brevity. We assume that reading a gating element costs c_r, the same as reading a primary element.

We seek to estimate a *shadow classifier* [2] $h_g : \mathcal{W} \times G \rightarrow$

[1] For simplicity, we assume here that classification cost is independent of the size of window being classified.This is true for sparse features such as Viola-Jones, but less so for dense ones such as HOG (Dalal, Triggs, and Schmid 2006). We also assume fully "power proportional" imagers, so that power consumed to read the imager is strictly proportional to the number of elements/pixels read. Extensions to, e.g., limited power proportionality and dense classifiers are straightforward.

[2] In line with traditional *sliding window* approaches, we model each window as independent of others. In reality, values of windows may be correlated, suggesting a structured output approach.

$\{0,1\} \times [0,1]$ along with threshold τ^*. h_g returns a prediction of h_p's output and a confidence of this prediction. The shadow classifier is used as part of a derived *hybrid* classifier h_{gp} ($x.i$ is the i'th component of x below):

$$h_{gp}(w,f,g) = \begin{cases} h_g(w,g).1, & \text{if } h_g(w,g).2 > \tau^* \\ h_p(w,f), & \text{otherwise} \end{cases}$$

In other words, for each window, the hybrid classifier returns the result of the shadow classifier on the gating imager if it is confident enough in it, and otherwise falls back on the primary classifier applied to the primary imager. To control costs, we budget a fraction $\lambda^* \ll 1$ of all windows to be processed by the primary classifier; beyond this budget, the hybrid classifier returns 0 (to indicate no object detected) on remaining windows w in the frame. Let n_{g_t} be the number of gating elements read processing frame f_t, and say applying h_g to classify window w using gating frame g_t cost $c'_c(w, g_t)$. Then the cost of processing a frame f_t with the hybrid classifier is:

$$C_{t,h_g,h_p} \leq \sum_{w \in W_t} c'_c(w, g_t) + n_{g_t} c_r + \lambda^* C_p \tag{1}$$

Given that λ^* is small and $n_{g_t} < \sum_i m_i * n_i \ll mn$, as long as (i) n_{g_t} is small and (ii) the expected cost of gating computations, $E_{w,g_t}[c'_c]$ is significantly smaller than the primary classification cost c_c, gating should yield significant savings.

The ideal h_g would minimize the disparity between its output and that of the primary classifier on training data ($F = \{f_t\}, G = \{g_t\}$), while staying, on average, within budget. We do not currently seek to track temporal correlations across frames, so that (f_t, g_t) pairs are assumed to be iid. We use a generic loss function $\mathcal{L}$ below; most standard loss-functions that yield confidence-reporting (e.g., margin-based or probabilistic) classifiers will suffice:

$$h_g = \operatorname*{argmin}_h \sum_{\substack{t \in 1\ldots T \\ w \in W^t}} \mathcal{L}(h, w, g_t, h_p(w, f_t)) \tag{2}$$
$$\text{subject to } \operatorname*{E}_t[C_{t,h,h_p}] < B$$

Algorithms

In the absence of the cost constraint, and if representative examples of F and G were available offline, Problem 2 would be a traditional learning problem with h_p providing labels for h_g. The constraint, however, requires us to favor classifiers that select gating elements that keep cost down. Further, we wish to assume that F and G are partly revealed in an online fashion, so that the (cost-sensitive) classifier needs to be re-estimated online. We discuss how to achieve these goals.

Adaptive Classification

We begin with incorporating the cost constraint. One approach (Turney 1995; Benbasat and Paradiso 2007; Xu et al. 2013) is to learn a decision tree that, when run, is structured to minimize gating sensor reads. It is unclear how to re-estimate these online. We therefore take a common alternate approach (Gao and Koller 2011; Yang et al. 2006; Zheng, Rish, and Beygelzimer 2005) of learning a classifier using a classification-cost-*in*sensitive objective function but applying this classifier in a cost-*sensitive* manner. This approach assumes a joint distribution $Pr(\mathcal{S}, \mathcal{Y})$, where $\mathcal{S}$ are random variables representing sensors and $\mathcal{Y}$ are r.v.'s representing values to be inferred, and a budget N of sensor readings. It iterates over N steps, at step i greedily selecting the sensor $S_i \in \mathcal{S} \setminus \mathcal{S}_{i-1}$ that maximizes the reduction in entropy (or intuitively, uncertainty) expected[3] when its value s_i is added to the values $\hat{\mathbf{s}}_{i-1}$ read from the sensors $\mathcal{S}_{i-1}$ selected so far:

$$\mathcal{S}_i = \mathcal{S}_{i-1} \cup \{ \operatorname*{argmax}_{S_i \in \mathcal{S} \setminus \mathcal{S}_{i-1}} \operatorname*{E}_{s_i \sim Pr(S_i|\hat{\mathbf{s}}_{i-1})} [-H(\mathcal{Y}|s_i, \hat{\mathbf{s}}_{i-1})] \} \tag{3}$$

If H changes slowly enough at any point, or if classification results are sufficiently skewed (i.e., $Pr(\mathcal{Y} = Y^*|\hat{\mathbf{s}}_i) - Pr(\mathcal{Y} = Y_j \in \text{dom}(\mathcal{Y}) \setminus \{Y^*\}|\hat{\mathbf{s}}_i) > \Delta_s$ for $Y^* = \text{argmax}_{Y \in \mathcal{Y}} Pr(Y|\hat{\mathbf{s}}_i)$ and threshold Δ_s), iteration may stop early, and the corresponding classification result, i.e., $E[Pr(\mathcal{Y}|\hat{\mathbf{s}}_i)]$ or Y^* respectively, returned. *Early stopping* of this kind short-circuits the overhead of sensor-selection optimization and can be critical for good performance.

Ensemble Adaptive Classification Applying the above perspective to our problem, we interpret each window-variable W^t_{ijs} as a random variable in $\{0,1\}$ and associate a r.v. $G_{ijmt} \in \mathbb{R}$ with each gating element g_{ijmt}, with $\mathcal{W}_t = \{W^t_{ijs}\}$ and $\mathcal{G}_t = \{G_{ijmt}\}$. We treat the $\mathcal{W}_t$ as independent of each other and model the joint relationship between each of them and all the gating r.v.'s using the Naive Bayes model: $Pr_W(W \in \mathcal{W}_t, \mathcal{G}_t) = Pr(W) \prod_{G \in \mathcal{G}_t} Pr(G|W)$ with models corresponding to W^t_{ijs} sharing parameters across time steps t. Finally, since the target variables are independent, the entropy reduction of the entire ensemble of models is the sum of the individual reduction of each one. An approximation that will turn out to be key to fast optimization is that when calculating expected entropy reduction, each model uses its own joint distribution Pr_W in order to calculate the distribution G_i of a candidate sensor conditioned on values seen $\hat{\mathbf{g}}_{i-1}$ so far:

$$G^*_i = \operatorname*{argmax}_{G_i \in \mathcal{G} \setminus \mathcal{G}_{i-1}} \sum_{\substack{W \in \\ \mathcal{W} \setminus \mathcal{W}_{i-1}}} \operatorname*{E}_{\substack{g_i \sim \\ Pr_W(G_i|\hat{\mathbf{g}}_{i-1})}} [-H(W|g_i, \hat{\mathbf{g}}_{i-1})] \tag{4}$$

It is unclear how to estimate a budget N for iterations i, since the cost of each iteration includes the cost of the sensor-selection optimization itself, in addition to reading and classification costs as in Equation 1. We therefore assume we can directly measure cost incurred (e.g., by measuring power or counting instructions executed) and halt iterating when we have just $\lambda^* C_p$ of our budget left or less. Finally, we use a posterior-skew based early stopping scheme. In step $i > 1$, if $|Pr(W = 1|\hat{\mathbf{g}}_i) - Pr(W = 0|\hat{\mathbf{g}}_i)| > \Delta_s$, we return $(w, Pr(W = w|\hat{\mathbf{g}}_i))$ (w maximizes the probability) early as

[3] Note only the *expectation*, not the value, of the candidate variable is required. The value is read only for the optimal candidate.

the classification result and confidence score. We add W to the set $\mathcal{W}_{i-1}$ of r.v.'s *retired* in the preceding steps. Retired r.v.'s are not considered in future steps.

Sequential optimization keeps the number n_{g_t} of gating sensors read low. Early stopping can reduce the cost of later steps. However, our scheme still requires $\Theta(|\mathcal{W}||\mathcal{G}|)$ expected-entropy calculations per frame *in the first step*. Even this will usually vastly exceed the $O(|\mathcal{W}|c_c)$ operations to simply invoke the primary classifier. We need optimizations to dramatically lessen both the *number* and *cost* of expected-entropy calculations.

Fewer Expected-Entropy Gain Calculations Notice that that as defined, *every* window is conditioned on *every* gating element G_I. This implies that the entropy term for every window will have to be re-evaluated for every gating variable at every step, an extremely expensive proposition. For most applications, it is likely that most windows are only determined by a small set of gating elements, e.g., their spatial neighbors. We therefore *prune* all models by removing all gating variables with low mutual information with respect to the window (below, $I(X;Y) = H(X) - H(X|Y)$):

$$Pr(W_{ijs}, \mathcal{G}) \mapsto Pr(W_{ijs}, \{G \in \mathcal{G} | I(W_{ijs}; G) > H_{\text{min}}\})$$

This restriction reduces expected-entropy calculations in the first step to $O(|\mathcal{W}|\mathcal{N}^{\mathcal{G}}_{\text{max}})$, where $N^{\mathcal{G}}_{\text{max}} \ll |\mathcal{G}|$ is the maximum number of gated variables any pruned model may contain. Further, in any step after the first, we only need to recompute $O(\mathcal{N}^{\mathcal{W}}_{\text{max}}\mathcal{N}^{\mathcal{G}}_{\text{max}})$ expected-entropy calculations, where $\mathcal{N}^{\mathcal{W}}_{\text{max}} \ll |\mathcal{W}|$ is the maximum number of windows any gated variable is associated with, since only those distributions Pr_W conditioned on the variable read in the previous step will change. For each of these $O(\mathcal{N}^{\mathcal{W}}_{\text{max}})$ distributions, we need to consider expectations over $O(\mathcal{N}^{\mathcal{G}}_{\text{max}})$ gating variables conditioned additionally on the newly observed value, yielding the above $O(\mathcal{N}^{\mathcal{W}}_{\text{max}}\mathcal{N}^{\mathcal{G}}_{\text{max}})$ bound.

We now focus on reducing the cost of the first step. In the $i = 1$ step, $\hat{\mathbf{g}}_{i-1}$, $\mathcal{W}_{i-1}$ and $\mathcal{G}_{i-1}$ are all empty so that the expected-entropy calculation of Equation 4 becomes simply:

$$G^*_1 = \operatorname*{argmax}_{G_i \in \mathcal{G}} \sum_{W \in \mathcal{W}} E_{g_i \sim Pr_W(G_i)}[-H(W|g_i)] \qquad (5)$$

Since this calculation does not depend on observations from each frame, we can *pre-compute* it once when $Pr(\mathcal{W}, \mathcal{G})$ is defined and start every frame with pre-initialized expected-entropy values for every window-variable and a fixed initial gating variable G^*_1 to read. This optimization drives the overhead of the first step to zero.

Cheaper Expected-Entropy Gain Calculations Consider the calculations in the remaining steps $i > 1$.

The outermost calculation is the maximization over G_i's of total expected gain over all windows if G_i were observed, conditioned on sensors read so far. Note that because of pruning, at the end of each step i, the total expected gain for most of the G_is remains unchanged, because the variable G^*_i chosen to be read is independent of them. We maintain a list of all G_i sorted by total expected gain if it were observed, given readings so far. The list is re-initialized at every frame to the value pre-calculated when $P(\mathcal{W}, \mathcal{G})$ is estimated as per Equation 5. At the end of every step i, we update the position in this list of just the gating variables affected by performing the $O(\mathcal{N}^{\mathcal{W}}_{\text{max}}\mathcal{N}^{\mathcal{G}}_{\text{max}})$ expected-entropy calculations triggered by reading G^*_i. Since the number of affected variables is usually small, the maximization in Equation 4 is fast.

The remaining calculation is that of the expectation of the entropy over the conditional distribution $Pr_W(G_i|\hat{\mathbf{g}}_{i-1})$, the $E[\ldots]$ term of Equation 4:

$$\mathop{\mathrm{E}}_{\substack{g_i \sim \\ Pr_W(G_i| \\ \hat{\mathbf{g}}_{i-1})}} [- \mathop{\mathrm{E}}_{\substack{w \sim \\ Pr_W(W| \\ g_i, \hat{\mathbf{g}}_{i-1})}} [\log Pr_W(W|g_i, \hat{\mathbf{g}}_{i-1})]] \qquad (6)$$

Since the Ws are Booleans (indicating presence or absence of the object), the inner expectation is dominated by the cost of calculating the log-probability. The latter can be minimized by a combination of using look-up tables to implement logs, and incremental recalculation of the product $Pr_W(W|g_i, \hat{\mathbf{g}}_{i-1}) \propto \prod_{g \in g_i, \hat{\mathbf{g}}_{i-1}} Pr_W(g|W)$. The bigger opportunity, however is in eliminating many of these innermost calculations altogether. To this end, note that every window W currently has a distinct model Pr_W. But because object models are translationally invariant for the most part[4], many windows should be able to share the same model, if the models are defined relative to their neighboring gating variables. We therefore index gating variables in a model by their spatial location relative to their primary imager window and tie parameters of "similar" models as follows.

We represent models as vectors of their parameters $(Pr(W_{ijs}), \ldots, Pr(G_{\kappa_{ijs}(k_q,l_q,m_q)}|W_{ijs}), \ldots)$, where $\kappa_{ijs}(k,l,m) = (k', l', m')$ such that (k', l', m') are the *relative* coordinates of gating element (k, l, m) in imager γ_m from the center of W_{ijs}. We cluster these vectors by L_2 distance using a simple cross-validated k-means scheme, and tie all corresponding parameters in a single cluster to their median value in the cluster; say $\kappa(W)$ is the representative model for window W. The log-probability calculation for all models in a cluster now becomes $\log Pr_{\kappa(W)}(W|g_i, \hat{\mathbf{g}}_{i-1})$. Since very many windows share values of $\hat{\mathbf{g}}_1$ and even $\hat{\mathbf{g}}_2$ (since, e.g., many background pixels have the same temperature), we can cache the value of the log-probability calculation to great effect.

The cost of evaluating the outer expectation depends on the representation of conditionals $Pr_W(G|W)$. For discretized representations, cost is proportional to $|\text{dom}(G)|$, and for continuous ones, to the cost of the integral. We represent the conditionals as Gaussians and use standard adaptive discretization or quadrature-based integration to cut costs in these settings. However, we stand to gain much more if we replace the variable G_i with $\text{dom}(G_i) \in \mathbb{R}$ with a *decision stump* $G_{i<}$ with $\text{dom}(G_{i<}) \in \{0, 1\}$ (such that $g_{i<} = 0$ if $g_i < \tau_i$ and 1 otherwise for some *decision threshold* $\tau_i \in \mathbb{R}$): in this case, $|\text{dom}(G)| = 2$! Ideally, the $G_{i<}$ should predict W as well as G_i. In practice, we choose τ_i such that $Pr(W = 1|g_i < \tau_i) \leq 0.01$ and $Pr(W = 1|g_i \geq \tau_i) \geq 0.3$. In other words, we want a

[4] Although, e.g., objects may have higher prior probabilities toward the middle of the image and lower toward the top or bottom.

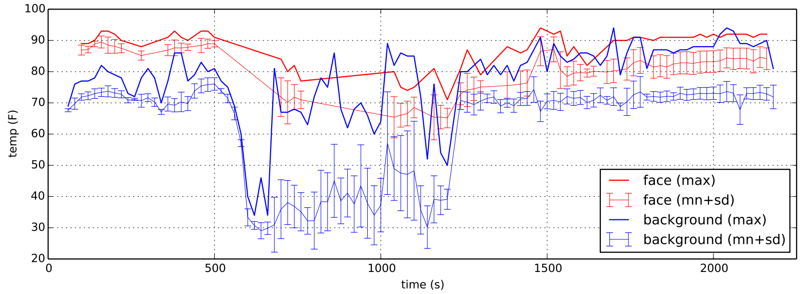

Figure 2: Variation of face vs background temperature over a 37-minute walk

very high-recall threshold that still has modest precision. If such a τ_i is unavailable, we use the original r.v. G_i.

Online Re-estimation

We now turn to online estimation of the classifier and the sensor selection policy, which are both controlled by the joint distributions $Pr(W \in \mathcal{W}, \mathcal{G})$. In our setting, the central issue is that these distributions may need to change over time due, e.g., to background temperature changes across locations or seasons, and face temperature differences across people in different settings. Since the shadow classifier only seeks to mirror the primary classifier, we can, in principle sample the primary classifier and gating sensors jointly to empirically estimate $Pr(W, \mathcal{G})$.

Assuming uniform sampling of Ws, the maximum likelihood estimate for $Pr(G \in \mathcal{G} = g|W = w)$ is simply $\#(W, w, G, g)/\#(W, w)$, where $\#(W, w, G, g)$ is the number of times the classifier reported value w for W when sensor G reported g and $\#(W, w)$ is the total number of times sensor W reported w. Given the additional cost of reading sensors, we face two issues. First, how do we trade off our budget on exploration (e.g., sampling windows for objects) versus exploitation (using the estimated model as in the previous sub-section for cost-sensitive classification)? Second, how do we trade off the benefit of applying a more current estimate of the joint with the cost of re-optimizing the ensemble models (e.g., model-pruning, pre-computing $i = 1$ expected-entropy numbers, model clustering and domain-collapsing via decision stumps) when applying it?

We address the latter problem first. Essentially, we *apply* newly estimated models when they differ substantially from existing models. For each representative model Pr_κ currently in use, if its latest estimate is Pr'_κ, we replace Pr_κ with Pr'_κ when the "distance" between the two distributions exceeds an empirically-determined threshold D_R. We use the Kullback-Leibler divergence $D(Pr_\kappa \parallel Pr'_\kappa) = E_{Pr_\kappa}[\log Pr_\kappa/Pr'_\kappa]$ as the measure of distance. When replacing the old model with the new, we preserve the pruning and clustering structure of the old, mainly in order to avoid the cost of doing so and because this structure tends to remain unchanged for given object/sensor-ensemble pairs. We do re-compute the $i = 1$ expected-entropy values and decision stumps, however. This situation is not entirely satisfactory. We anticipate future work that will perform pruning, clustering and other optimizations incrementally and efficiently and provide solid theoretical grounding (e.g., via regret bounds) on the design choices.

We address the exploration-exploitation trade-off via a simple ϵ-greedy (Watkins 1989) approach. At every time step i, of the adaptive classification loop, we either read a (W, G) pair sampled from $\mathcal{W} \times \mathcal{G}$ and use it to update $\#W$,$\#(W, w, G, g)$ and $\#(W, w)$ with probability $\epsilon \ll 1$ or, with probability $1 - \epsilon$, compute G_i^* as usual per Equation 4. We choose ϵ to incur cost sufficient to read and classify a few imager windows per frame: for small n, $\epsilon = nC_p/|W^t|$.

An important detail is that unlike conventional ϵ-greedy schemes, sampling $\mathcal{W} \times \mathcal{G}$ *uniformly* may be inadequate because objects to be detected are rare. Uniform sampling produces an excellent estimate for $Pr(G|W = 0)$ quickly, but a poor one for $Pr(G|W = 1)$. We therefore first uniformly sample $g \sim G$, but only sample W gated by G if $g > g^*$, where g^* is a crude empirically determined bound such that $Pr(W = 1|G > g^*)$ is not close to zero. For instance, for face detection, a good value for g^* is the 95th percentile of the background temperature distribution. Further, if we do find a face, instead of updating $Pr(G|W)$ for the particular G we sampled, we also update $Pr(G_i|W)$ for all G_i in models containing W and G_i.

Evaluation

We seek to answer two questions: Can our system yield substantial speedups over non-adaptive or offline systems? How much do individual optimizations contribute?

We collected QVGA (320x240) far-infrared video at 10fps with aligned VGA (640x480) RGB frames of daily life in three settings: "office", "walk" and "lobby". The data was collected over a period of a six weeks of winter and spring. We downsampled the video to produce an ensemble of resolutions $64 \times 48, 32 \times 24, 16 \times 12, 8 \times 6$ and 4×3. Each video collection session lasted 40-60 minutes, with a total 10hrs of data for each scenario. In every scenario, the camera was held to point forward at chest height (as a wearable would). The "office" sessions were shot entirely within a single large office building, representing typical footage from an office worker, with almost all faces at steady temperature. The "walk" sessions were between buildings, usually with tens of minutes between buildings. The "lobby" setting has a mix of people entering from the outside, long-time indoor residents and those who are warming up.

Figure 2 shows mean, maximum and standard deviation of face and background temperatures during a walk, with a reading shown every 20s. Faces are marked with red whiskers. The camera is in an office until the 500s mark, outdoors until 1200s and in a crowded lobby after. Note that

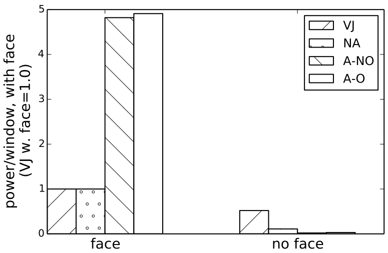

Figure 3: Impact of gating on power consumption.

face and background are usually well separated (as per the gap between standard deviation whiskers), although it not uncommon (usually under 5%) for a few pixels of background to mimic face temperature (thus arguing for adaptive tests beyond temperature thresholding). Further, the separating temperature changes with location (lower outdoors in this case) and with mix of people in field of view (lobby has a range of face temperatures), favoring online re-estimation.

We run Viola-Jones face detection to detect faces in our data. We instrument Viola-Jones to count the number of pixels p read and number of instructions i (adds, multiplies and compares) executed per frame. Similarly, we run our Ensemble Adaptive Classification framework on the same data and count instructions and gating pixels read. As per the state of the art (Aptina 2011; Hori and Kuroda 2007; Gerosa et al. 2009), we attribute $P = 40nJ$ to read a pixel and $I = 5nJ$ per instruction for a total estimate of $pP + iI$.

Figure 3 shows the average power consumed to process a window containing a face (left cluster) and no face (right) using RGB only (labeled VJ), non-adaptive FIR gating (NA), adaptive gating with no online re-estimation (A-NO) and adaptive gating with online re-estimation (A-O). A-O and A-NO are trained on "office" and "lobby" data but not "walk" data to gauge if online estimation develops a better model for "walk". In NA, we look for gating pixels with temperature t_{NA} over the 95th percentile of observed temperatures, estimated over the past 5 frames. If a qualifying pixel is seen, NA searches a neighborhood of this pixel in the primary imager using VJ. The figure shows costs relative to VJ. The "face" bars for A-O and A-NO include the cost of VJ, since VJ is invoked when these detectors detect a face.

Some points are worth noting. First, when a face is in the window, adaptive techniques cost noticeably more than VJ. They execute multiple steps of expected entropy optimization. As implemented, steps beyond $i = 1$ are relatively inefficient especially compared to VJ. On the other hand, NA compares a gating pixel to t_{NA} and invokes VJ, so it costs almost the same as VJ. Second, with no face in the window, adaptive techniques win big (1 and 3% of VJ): given the *pre-computing* and *decision stump* optimizations, most gating pixels G are simply compared, classified as part of models $Pr_{\kappa}(W|G)$ and retire several windows W as likely not faces. NA conservatively calls VJ on roughly 5% of these windows, incurring a noticeable cost. Third, the cost of VJ is still relatively high with no faces in the frame; even though its cascades short-cut out, it must read the whole primary imager. **Overall, given that below 1% of windows contain faces, we estimate $\sim 50\times$ total savings for A-O/NO.**

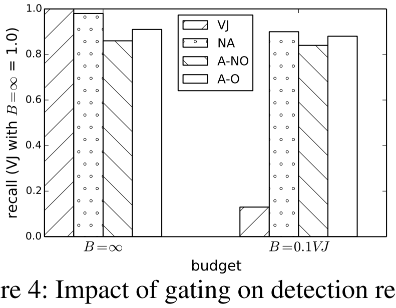

Figure 4: Impact of gating on detection recall.

Optimization dropped	Power consumed (relative to all optimizations)
none	1.0
pruning	6412
precomputing	18.1
clustering	2.2
dec. stumps	4.2

Table 1: Impact of dropping optimizations on power.

Figure 4 shows that the gains in power are not at the expense of accuracy. As a baseline (B_{∞}), we measure the fraction of faces found by the standard VJ algorithm that were also found by NA, A-O and NO. In this case, A-NO and A-O find 86 and 91% of these faces. Many of the additional faces missed by A-NO were in the "walk" dataset; **online estimation provides a small but noticeable boost**. In a more realistic setting, we limited the budget to $B = 0.1VJ$, 10% of the average cost per frame of applying VJ. We now modify all algorithms to stop processing windows and report 0 (i.e., "no face") if they exceed their budget. The performance of VJ now collapses while the others maintain performance.

Table 1 lists the effect on power savings of turning off each key optimization. Not pruning, i.e., not allowing every window to depend on every gating pixel, unsurprisingly is very beneficial; entropy-based pruning cures the problem quite well. Second, **pre-computing the $i = 1$ step and representing $Pr(G|Y)$ with decision stump $Pr(G_{>}|Y)$ are critical to performance because they allow most windows to retire with a single compare and multiply as above**. Model-clustering is important not only in the $i \geq 2$ steps, but also to control the cost of online estimation.

Conclusions

We have introduced a new problem, *Ensemble Shadow Classifier Learning* of budgeted adaptive classification and online re-estimation for sensor arrays. The core challenge in such systems is to retain the efficiency of decision-tree style classifiers while allowing online re-estimation as in Bayesian value-of-information (VOI) based systems. We have introduced a suite of optimizations to bridge this performance gap while adding an online component to the VOI setting. Early measurements show substantial performance gains. We anticipate the extension of recent formal results on online adaptive submodular optimization to this setting.

References

Aptina. 2011. 640h x 480v, ultra low-power cmos digital image sensor.

Benbasat, A. Y., and Paradiso, J. A. 2007. A framework for the automated generation of power-efficient classifiers for embedded sensor nodes. In *SenSys*, 219–232.

Buddharaju, P.; Pavlidis, I.; Tsiamyrtzis, P.; and Bazakos, M. 2007. Physiology-based face recognition in the thermal infrared spectrum. *IEEE Trans. Pattern Anal. Mach. Intell.* 29(4):613–626.

Dalal, N.; Triggs, B.; and Schmid, C. 2006. Human detection using oriented histograms of flow and appearance. In *ECCV (2)*, 428–441.

Dereniak, E. L., and Boreman, G. D. 1996. *Infared Detectors and Systems*. Wiley, second edition.

Gabillon, V.; Kveton, B.; Wen, Z.; Eriksson, B.; and Muthukrishnan, S. 2013. Adaptive submodular maximization in bandit setting. In *NIPS*, 2697–2705.

Gao, T., and Koller, D. 2011. Active classification based on value of classifier. In *NIPS*, 1062–1070.

Gerosa, G.; Curtis, S.; D'Addeo, M.; Jiang, B.; Kuttanna, B.; Merchant, F.; Patel, B.; Taufique, M.; and Samarchi, H. 2009. A sub-2 w low power ia processor for mobile internet devices in 45 nm high-k metal gate cmos. *Solid-State Circuits, IEEE Journal of* 44(1):73–82.

Golovin, D., and Krause, A. 2011. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *J. Artif. Intell. Res. (JAIR)* 42:427–486.

Greiner, R.; Grove, A. J.; and Roth, D. 2002. Learning cost-sensitive active classifiers. *Artif. Intell. J.* 139(2):137–174.

Hori, Y., and Kuroda, T. 2007. 0.79 mm^2 29-mw real-time face-detection core. In *JSSC*.

Howard, R. 1966. Information value theory. *Systems Science and Cybernetics, IEEE Transactions on* 2(1):22–26.

LiKamWa, R.; Priyantha, B.; Philipose, M.; Zhong, L.; and Bahl, P. 2013. Energy characterization and optimization of image sensing toward continuous mobile vision. In *MobiSys*, 69–82.

Socolinsky, D. A., and Selinger, A. 2004. Thermal face recognition in an operational scenario. *2013 IEEE Conference on Computer Vision and Pattern Recognition* 2:1012–1019.

Streeter, M. J., and Golovin, D. 2008. An online algorithm for maximizing submodular functions. In *NIPS*, 1577–1584.

Turney, P. D. 1995. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *J. Artif. Intell. Res. (JAIR)* 2:369–409.

Viola, P. A., and Jones, M. J. 2004. Robust real-time face detection. *International Journal of Computer Vision* 57(2):137–154.

Watkins, C. J. C. H. 1989. *Learning from delayed rewards*. Ph.D. Dissertation, U./ of Cambridge.

Wolff, L.; Socolinsky, D.; and Eveland, C. 2005. Face recognition in the thermal infrared. In Bhanu, B., and Pavlidis, I., eds., *Computer Vision Beyond the Visible Spectrum*, Advances in Pattern Recognition. Springer London. 167–191.

Xu, Z.; Kusner, M.; Chen, M.; and Weinberger, K. Q. 2013. Cost-sensitive tree of classifiers. In Dasgupta, S., and Mcallester, D., eds., *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, 133–141. JMLR Workshop and Conference Proceedings.

Yang, Q.; Ling, C. X.; Chai, X.; and Pan, R. 2006. Test-cost sensitive classification on data with missing values. *IEEE Trans. Knowl. Data Eng.* 18(5):626–638.

Zhang, L.; Wu, B.; and Nevatia, R. 2007. Pedestrian detection in infrared images based on local shape features. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 1–8.

Zheng, A. X.; Rish, I.; and Beygelzimer, A. 2005. Efficient test selection in active diagnosis via entropy approximation. In *UAI*, 675–.

Zubek, V. B., and Dietterich, T. G. 2002. Pruning improves heuristic search for cost-sensitive learning. In *ICML*, 19–26.