

Class-Specific Hough Forests for Object Detection

Juergen Gall
BIWI, ETH Zürich and MPI Informatik*
gall@vision.ee.ethz.ch

Victor Lempitsky
Microsoft Research Cambridge
victlem@microsoft.com

用于目标检测的特定类别霍夫森林

随机森林：

测试样本送入森林中，接受所有树的检验，对于每个树，该样本抵达的那个叶子作为输出结果的一项记录，所有记录的平均值作为该样本最终的输出。

构造霍夫森林：

训练树的过程中，图像块抵达某个节点的信息被储存下来，具体的，我们储存了抵达该节点的图像块集之中**源于目标图片的图块**所占的比重**C**以及偏移向量**D**（图块中心和原图中心的位置偏移向量，负样本中的图块的**D**为0）

图块外观和二值测试：

训练过程中，对于**树上的每一个非叶子节点**，都要针对图块**I_i**的表现进行一个二值测试。训练和测试的时候，所有图块都是固定大小，如16*16pixel，图块**I_i**的表现由提取到的特征通道定义，即**I_i={I_{i1}, I_{i2}, ..., I_{iC}}**，**C**是通道数。

The binary tests on a patch appearance $t(I) \rightarrow \{0, 1\}$ can be defined in several ways, but we have chosen simple pixel-based tests. Such a test is defined by a channel $a \in \{1, 2, \dots, C\}$, two positions (p, q) and (r, s) in the 16-by-16 image, and a real handicap value τ . The test $t_{a,p,q,r,s,\tau}(I)$ is then defined as:

$$t_{a,p,q,r,s,\tau}(I) = \begin{cases} 0, & \text{if } I^a(p, q) < I^a(r, s) + \tau \\ 1, & \text{otherwise.} \end{cases} \quad (1)$$

树的构建：

从根节点递归地构造树——构建过程中，每个节点储存了一个训练图块集，**如果该节点深度达到最大深度（D_{max}=15）或者节点储存的图块集数量很小（M_{min}=20），则停止递归，令该节点为叶子节点**，并且将叶子的投票信息（**C_L**, **D_L**）积累并存储起来。**否则，该节点被创建为非叶子**，并从一个大的随机生成的二值测试池选择一个最优的二值测试。根据所选的二值测试，将到达该节点的训练图块集分为2个子集（输出0的为一个子集，输出1的为另外一个子集），分别成为该节点的2个子节点。

训练建树的图示步骤：

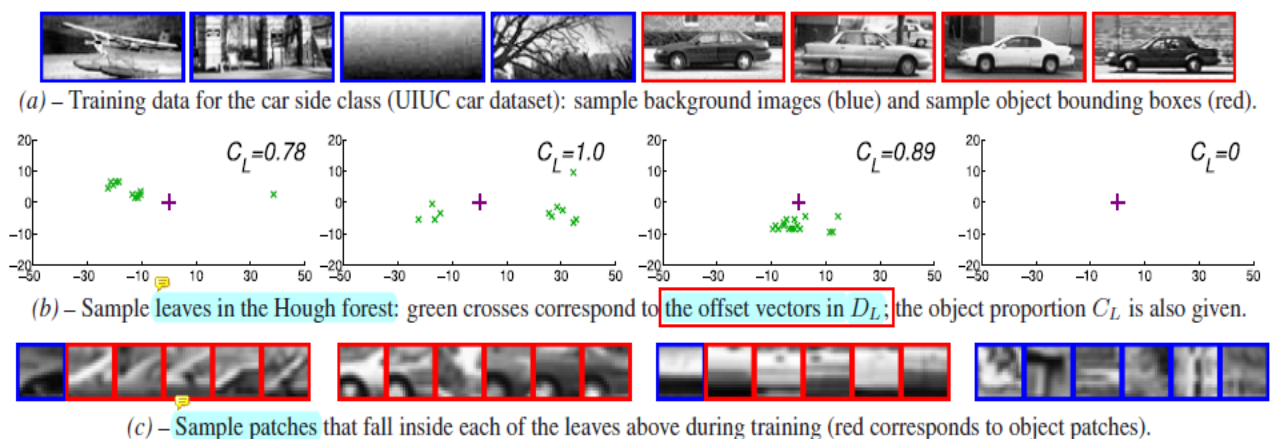


Figure 2. For the set of training images shown in (a), we visualize the data recorded in some of the leaves of the constructed class-specific Hough forest in (b). This data consists of the object patch proportion C_L and the list of the offset vectors for object patches D_L . Note that the leaves of the Hough forest form a discriminative class-specific codebook as shown in (c): the training examples falling inside each of the first three leaves can be associated with different parts of a car.

(b) 每个叶子存放了落入该叶子的图块集中**目标子区域的比重**，以及各目标子区域中心**偏移向量**数组。

(c) 对应上图中各个叶子的图块集中的，部分示意图块图。注意到霍夫森林中的所有叶子构成了一个区别力很强的特定类别分类的编码本：那些落入前三个叶子中的训练样本可以认为是汽车目标的不同部分。

构建霍夫森林的关键在于，二值测试评测的质量如何。

类的比重**C_L**和偏移向量列表**D_L**都用于投票。

一个更高级的方法是去选择二值测试方法，使得分类标签和偏移向量的不确定性，沿着树枝不断减少。

set of patches $A = \{\mathcal{P}_i = (\mathcal{I}_i, c_i, d_i)\}$. The **class-label uncertainty** measuring the impurity of the class labels c_i is defined as:

$$U_1(A) = |A| \cdot Entropy(\{c_i\}), \quad (2)$$

where the entropy of the set of binary values $\{c_i\}$ with the mean value $c \in [0, 1]$ is defined in a standard way: $Entropy(\{c_i\}) = -c \cdot \log c - (1-c) \cdot \log (1-c)$. The second measure called the **offset uncertainty** and corresponding to the impurity of the offset vectors d_i is defined simply as:

$$U_2(A) = \sum_{i:c_i=1} (d_i - d_A)^2, \quad (3)$$

where d_A is the mean offset vector over all object patches in the set². Note that the background patches corresponding to $c_i = 0$ are ignored here.

如何选择二值测试的函数呢？

1. 先均匀的采样a, p, q, r, s, t生成一个二值测试池 t^k
2. 假设数据的真实差值为100, 则t (差值T) 取为50-100之间的均匀随机数
3. 然后随机的选择优化方式, 即该节点可以最小化类别不确定度, 也可以最小化偏移不确定度。
4. 一旦负样本区块总数少于5%, 则节点进行优化的方式为——最小化偏移不确定度。
5. 最终选择这样的二值函数 t^k ——它把所有图块分为2部分后, 2个图块子集各自的不确定度总和最小。

the binary test with the minimal sum of the respective uncertainty measures for the two subsets, it splits the training set into:

$$\operatorname{argmin}_k \left(U_{\star}(\{p_i | t^k(\mathcal{I}_i)=0\}) + U_{\star}(\{p_i | t^k(\mathcal{I}_i)=1\}) \right) \quad (4)$$

where $\star = 1$ or 2 (depending on our random choice).

减少类别标签不确定度的节点和减少偏移不确定度的节点之间进行交叉来建树, 这种树构建的过程保证了所有到达叶子的区块在类别标签和偏移属性上均具有较好的不变性。

结果便是, 霍夫森林的叶子们, 可以对目标的质心位置进行比较精准的投票。可用于目标位置检测了

使用霍夫森林进行目标检测:

假设在训练和测试的环节中, 所有目标边框均为 $W \times H$ 固定尺寸。则只需求出目标中心的位置, 即可确定目标整体的位置。

考虑测试图片中的任意图块, $\mathcal{P}(y) = (\mathcal{I}(y), c(y), d(y))$, y 是图块的中心。

$\mathcal{I}(y)$ 是图块表现, $c(y)$ 是未知的图块类别 (是否属于目标区域的部分, 是 $c(y)=1$, 否则 $c(y)=0$), $d(y)$ 是图块 $\mathcal{I}(y)$ 到测试图片中心的偏移 ($c(y)=1$ 才有效, 否则 $d=0$)。

$E(x)$ 表示目标中心位于测试图片的 x 处这一事件。

感兴趣的是, 计算条件概率 $p(E(x) | \mathcal{I}(y))$, $p(E(x) | \mathcal{I}(y))$ 表示图块 $\mathcal{I}(y)$ 所带来的目标位置估计——目标中心位于原图中 x 位置的概率。

分为两种情况: 图块中心 y 在目标边框 $B(x)$ 内, 或是目标边框外。

若 y 不在 $B(x)$ 内—— $\mathcal{I}(y)$ 不是目标的子图块, 则假设 $\mathcal{I}(y)$ 不贡献任何信息给 $E(x)$ 。 $p(E(x) | \mathcal{I}(y)) = p(E(x))$. 这里没有考虑到上下文的信息有效性。

我们仅考虑对目标检测有效的图块全部来自于目标区块的子区块。集中考虑 y 在 $B(x)$ 之中的情况。

一旦 y 在目标边框内, 则显然有 $c(y)=1$ —— $\mathcal{I}(y)$ 是目标的子区块。条件概率为:

$$\begin{aligned} p(E(x) | \mathcal{I}(y)) &= p(E(x), c(y)=1 | \mathcal{I}(y)) = \\ &= p(E(x) | c(y)=1, \mathcal{I}(y)) \cdot p(c(y)=1 | \mathcal{I}(y)) = \\ &= p(d(y) = y - x | c(y)=1, \mathcal{I}(y)) \cdot p(c(y)=1 | \mathcal{I}(y)). \end{aligned} \quad (5)$$

将区块 $\mathcal{I}(y)$ 传入霍夫森林的树之中, 可以得到 (5) 中的各个因式的值。假设在树 T 上, $\mathcal{I}(y)$ 落入叶子 L 上, 则第一个因式可由核密度估计 (Parzen窗) 得到, 第二个因式即为 C_L 。

单棵树的情况下, 该条件概率为核密度估计乘以 C_L 。——高斯核密度估计。

$$\begin{aligned} p(E(x) | \mathcal{I}(y); T) &= \\ &= \left[\frac{1}{|D_L|} \sum_{d \in D_L} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|(y-x)-d\|^2}{2\sigma^2}\right) \right] \cdot C_L \end{aligned} \quad (6)$$

对于多树森林, 我们将各个树的输出取平均值得到森林的输出。

$$p(E(\mathbf{x})|\mathcal{I}(\mathbf{y});\{\mathcal{T}_t\}_{t=1}^T) = \frac{1}{T} \sum_{t=1}^T p(E(\mathbf{x})|\mathcal{I}(\mathbf{y});\mathcal{T}_t). \quad (7)$$

以上2式表示了单个图块对于估计目标位置的贡献，即目标位置以该概率趋近于 \mathbf{x} 位置。

怎样进行各图块的投票呢？

将所有图块对应的条件概率累积起来，得到一副霍夫图像 $V(\mathbf{x})$ ——即对每一个像素位置，均得到此处作为目标中心的可能性估计。

$$V(\mathbf{x}) = \sum_{\mathbf{y} \in B(\mathbf{x})} p(E(\mathbf{x})|\mathcal{I}(\mathbf{y});\{\mathcal{T}_t\}_{t=1}^T). \quad (8)$$

找到最大的响应位置，即可认为此处为目标中心。