

我这周主要在看 Opencv 自带的 HOG 实现的源码，它实现的主要流程如下：首先计算检测图片可缩放的比例，然后开启多线程处理各个比例的图片的计算工作，在对一个图片进行检测的时候，使用 HOGCache 这个 HOG 存储结构对 64*128 的区域计算梯度等操作（每个检测窗口包含 105 个 block 且每个 block 包含 36 个 bin），使用这样的检测窗口对图片进行扫描检测，最后将检测结果汇总。

看了 Opencv 自带的 HOG 源码后发现它的实现与原论文的描述有些出入。例如，在 Opencv 的实现里会对检测区域加上一个 padding 再进行检测，此外，在计算过程中还会出现负坐标。总的来说，对代码的实现看得不是很懂，有些地方甚至一头雾水。

另外，我还找到《Fast Training of Pose Detectors in the Fourier Domain》这篇论文，它提供了一种能加快图片检测速度的方法。像 HOG 这类使用检测窗口对图片进行扫描检测的算法，相邻两次扫描的区域的数据大部分是相同的，因此存在很多冗余计算。而论文提出的方法在傅里叶域上减少这些冗余，进而提高算法的处理速度。因此我会在下周仔细研读这篇论文。