

[About](#) [Project](#) [Resume](#) [Blog](#) [Book](#) [Times](#) [Github](#)

图像检索：拓展查询(Query Expansion)

📅 2017年02月05日 📁 Image Retrieval 💡 拓展查询 字数:3581

拓展查询(QE, Query Expansion): 指对返回的前top@K个结果，包括查询样本本身，对它们的特征求和取平均，再做一次查询，此过程称为拓展查询。

从上面的定义可以看出，拓展查询属于重排的一种方式。通过Query Expansion，以达到提高检索召回率的目的。前面的博文[RANSAC算法做直线拟合](#)曾介绍过RANSAC的基本思想，放在词袋模型里（相应博文见[图像检索：BoW图像检索原理与实践](#)），我们可以使用RANSAC方法或Weak Geometry Consistency方法做几何校正，进行重排以提高检索的精度。在这篇博文中，小白菜暂时抛开其他的重排方法，重点分析Query Expansion对图像检索精度的提升。

根据小白菜读图像检索论文获得的对Query Expansion的感知，做完Query Expansion能够获得百分之几的精度提升。为了证实Query Expansion对检索精度的改善，在过去一段时间里，小白菜在Oxford Building数据库上对其做了验证。下面是小白菜对Query Expansion的实验整理和总结。

特征表达

Oxford Building图像数据库，每一幅图像提取的是一个512维的CNN特征，即对于Oxford Building图像数据库，我们得到5064个512维的特征。

特征索引

直接采用brute线性扫描，因为图库才5064张图像，所以没必要建索引。在实际应用中，我们可以采用哈希、倒排PQ等方式，这一部分可以细讲很多，有机会的话，小白菜单单独拿一个篇幅整理实用的索引方法。

评价指标

实验评价指标采用平均检索精度(mAP, mean average precision), mAP如何计算可以阅读[信息检索评价指标](#)，里面有对mAP如何计算的详细介绍。

对于Oxford Building图像数据库，mAP的计算过程有必要详细介绍一下。Oxford Building的groundtruth有三类：good, ok和junk。对于某个检索结果，如果它在good和ok中，则被判为是与查询图像相关的；如果在junk中，则被判为是不相关的。我们可以细致的阅读一下Oxford Building的mAP计算代码：

```
float compute_ap(const set<string>& pos, const set<string>& amb, const vector<string>& ranked_list){
    float old_recall = 0.0;
    float old_precision = 1.0;
    float ap = 0.0;

    size_t intersect_size = 0;
    size_t i = 0;
    size_t j = 0;
    for ( ; i<ranked_list.size(); ++i) {
        if (amb.count(ranked_list[i])) continue;
        if (pos.count(ranked_list[i])) intersect_size++;

        float recall = intersect_size / (float)pos.size();
        float precision = intersect_size / (j + 1.0);
```

```
    ap += (recall - old_recall)*((old_precision + precision)/2.0);

    old_recall = recall;
    old_precision = precision;
    j++;
}
return ap;
}
```

其中，`pos` 即是由good和ok构成的set，`amb` 是junk构成的set，`ranked_list` 即查询得到的结果。可以看到Oxford Building上计算的AP是检索准确率(precision)和检索召回率(recall)曲线围成的面积(梯形面积，积分思想)，mAP即是对AP的平均。

理解完了Oxford Building的mAP计算过程，还有一个需要考虑的问题是：对于查询图像特征的提取，我们要不要把Oxford Building提供的区域框用上，即在提取特征的时候，我们是在整个图像提取特征，还是在区域框内提取特征？在图像检索的论文中，在计算Oxford Building的mAP时，都是在区域框内提取特征。但是放在实际中，我们肯定是希望我们的图像检索方法能够尽可能的减少交互，即在不框选区域的时候，也能够取得很好的检索精度。所以，基于这样的意图，在实际中测评检索算法的mAP时，小白菜更喜欢采用在整个图像上提取特征的方式。当然，如果不嫌麻烦的话，可以两种方式都测评一下。

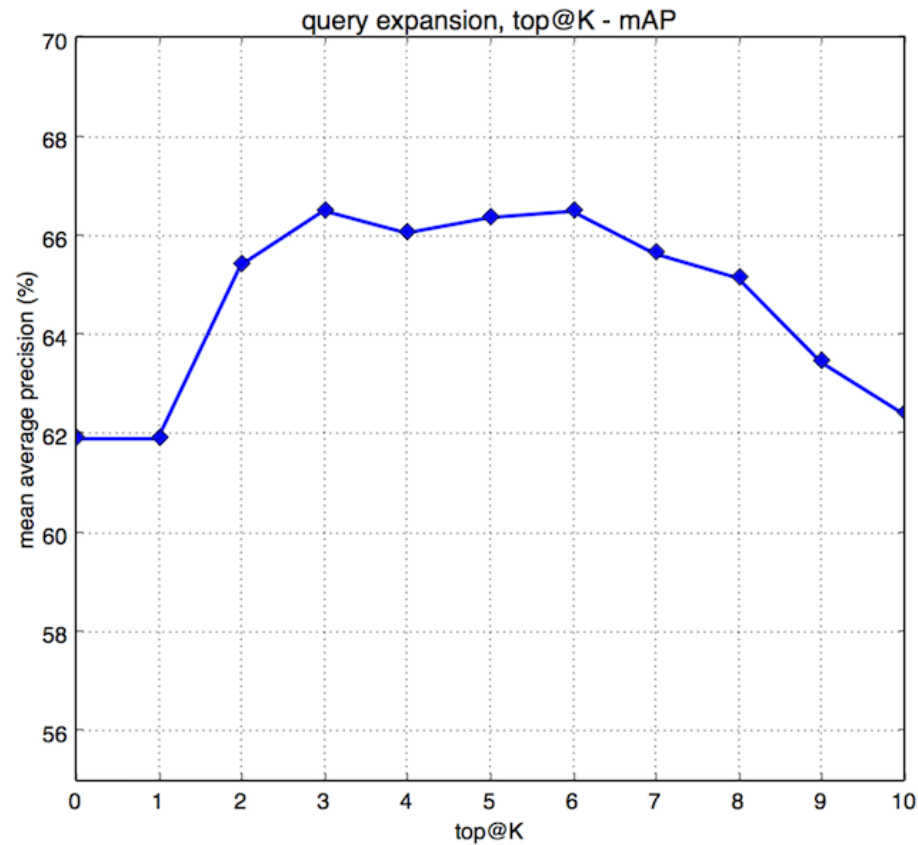
查询拓展对mAP的提升

库内查询，所以返回的top@1为查询图像自身，并且采用的是全图查询(即上面提到的对于查询图像是在整个图像上提取特征，而不是在区域框内提取特征)，表中top@K表示取前K个样本求和取平均。

top@K	0	1	2	3	4	5	6	7	8	9	10
-------	---	---	---	---	---	---	---	---	---	---	----

MAP	61.91%	61.91%	65.42%	66.52%	66.07%	66.38%	66.51%	65.65%	65.16%	63.46%	62.41%
-----	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

上面表格中mAP随top@K用曲线表示如下：



在不做Query Expansion的时候，即top@K=0时，mAP为61.91%。因为查询属于库内查询，所以top@K=1时，仍然是查询向量本身，故结果与top@K=0是一样的。从实验的结果可以看出，Query Expansion确实能够提升检索的精度，在top@K=3的时候，取得了最高的检索精度。相比于不做Query Expansion，Query Expansion可以提高4%-5%的检索精度。

所以，在实际中，做Query Expansion完全是有必要的，一则是它实现简单，二则是它提升的效果还是比较明显的

[← 机器视觉：Caffe Python接口多进程提取特征](#)

[图像检索：再叙ANN Search→](#)

comments powered by Disqus

Friend: Lichao [Yihui](#) [Xuezhi](#) [52ml](#) [cyq](#) [Rui Hu](#) [pyimagesearch](#) [bean](#) [ml dairy](#) [Resources](#)

Made with [Jekyll](#), hosted on [Github Pages](#). Inspired by [saunier](#), designed by [Willard](#).

[Attribution-NonCommercial-ShareAlike 4.0 International](#) 2013-2017