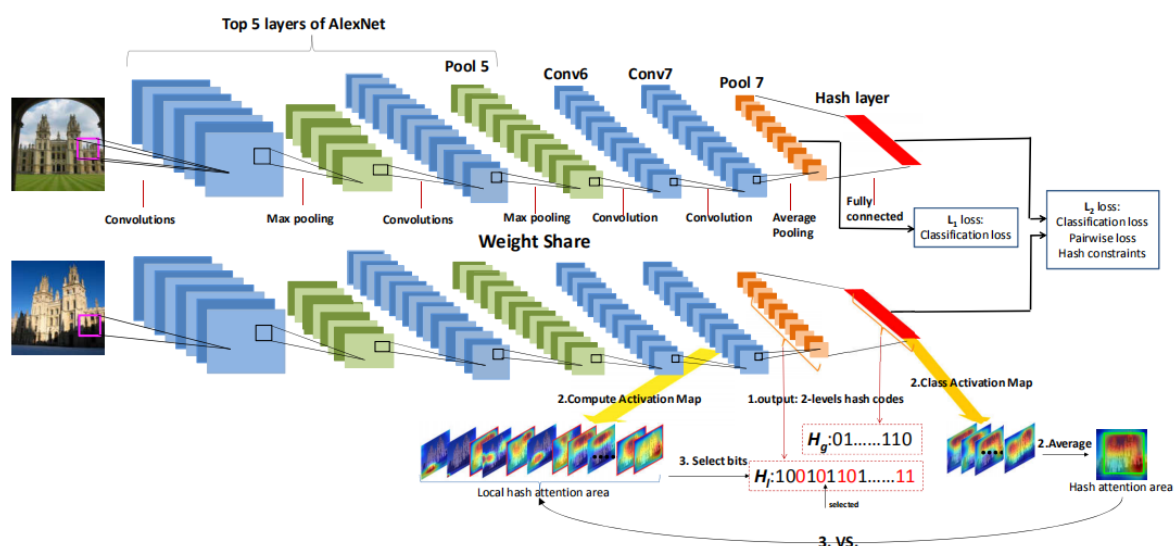


## 层级哈希

将单层哈希编码方案扩展为两级分层编码方案来增强哈希码的表现力，在该方案中，第一级哈希编码用于捕获图像的局部语义信息，而第二级则将其压缩为更具全局性的紧凑编码，然后通过分层方式组合两个级别的哈希编码，提高检索效率

一种依赖于激活图（activation map）和类激活图（class activation map, CAM）的注意力机制，该机制利用深度特征的局部和全局属性用以选择每个查询图像最重要的一些特定哈希编码位，避免使用第一级完整的哈希编码



CAM最初是用于获取同某语义类别相关的判别性图像区域，并不适合用于图像检索

我们将哈希编码的每一位视为一个类别，并使用CAM来定位同每个全局层级哈希编码位相关的图像显著区域。对于局部哈希编码，采用类似方法计算CAM

第一步：确定全局层级哈希码 $H_g$ 每一位在输入图像中的关注区域。首先，通过将每一编码位视为一个类别，计算它们类激活图 $M_1, M_2, \dots, M_H$ ，并对它们取平均获得 $M_{avg}$ 。然后，将该平均图 $M_{avg}$ 二值化为 $B_{avg} = \delta(M_{avg} > \theta)$ ，其中 $\theta$ 是阈值。最后，通过寻找 $B_{avg}$ 的最大连通子图确定关注区域。如图2.2（中间一栏）给出的所得平均二值激活图示例。

第二步：估计Conv7每个特征图的显著性得分。首先，使用文献[112]方法计算Conv7层所有特征图 $I_1, I_2, \dots, I_C$ 的激活图 $AM_1, AM_2, \dots, AM_C$ ，参见图2.2右图。然后，将这些激活图二值化为 $B_1, B_2, \dots, B_C$ 。最后，将每个特征图的显著性得分 $s$ 定义该特征图与平均特征图之间的重叠度，即：

$$s_i(B_i, B_{avg}) = \text{sum}(B_i \wedge B_{avg}) \quad (2.4)$$

其中 $\wedge$ 是逐位与运算， $\text{sum}$ 表示对矩阵所有元素求和。

第三步：选择 $H_l$ 的哈希编码位。首先，根据特征图 $I_1, I_2, \dots, I_C$ 的得分 $s$ 从高到低进行排序，选择前 $L$ 个作为信息最丰富的局部特征。然后选择 $H_l$ 对应的 $L$ 个比特（记为 $H'_l$ ）进行有效检索。这 $L$ 个选定索引将用于过滤每个被检索图像的第一级哈希编码，也就是说，只需要比较查询图像和每个图像 $i$ 的这些选定哈希位。形式上，令 $\Psi_q(*)$ 表示 $*$ 的 $L$ 位，这些 $L$ 位与查询哈希码 $H_q$ 中选择的 $L$ 位相同，有 $H'_q = \Psi_q(H_q)$ 。则查询哈希码与哈希码 $H_i$ 之间的汉明距离为：

$$d_H(H'_q, H_i) = d_H(H'_q, \Psi_q(H_i)) \quad (2.5)$$

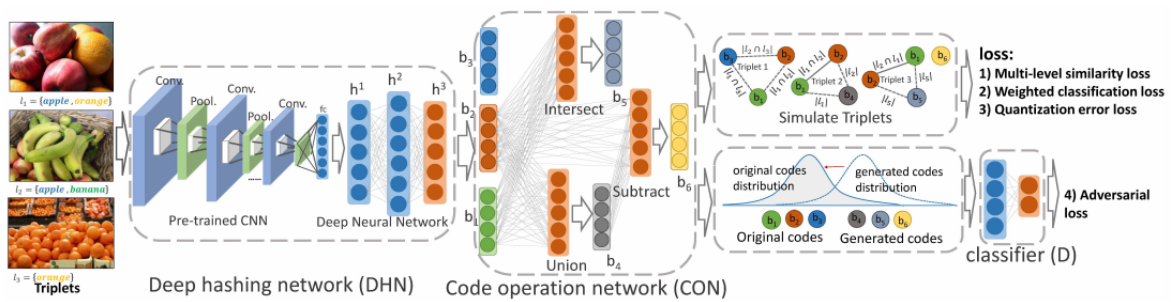
每一位编码对应的类激活图取平均，然后根据阈值二值化，确定关注区域

计算所有特征图的激活图，然后二值化，再根据关注区域计算重叠度

从高到低进行排序，选择前 $L$ 个作为局部特征，然后选择对应的 $L$ 位用于过滤被检索图像的第一级哈希编码

层次搜索策略，首先使用全局级别哈希编码进行粗检索，选出前 $K$ 个，再使用局部级别哈希编码进行重新排序

## 深度编码操作哈希（Deep Code Operation Hashing, DCOH）



$N$ 张图像， $C$ 个类别， $D$ 维视觉特征，学习 $K$ 个哈希函数，映射成 $K$ 维哈希编码

哈希编码操作符 分别为交 并 差， 对应得到的新的哈希编码的类别也是两个原始操作数类别的交 并 差

$$h_{x_1 \oplus 2} = f_{\oplus}(h(x_1), h(x_2)) \equiv W_{\oplus}[h(x_1), h(x_2)] \quad (3.3)$$

$$h_{x_1 \otimes 2} = f_{\otimes}(h(x_1), h(x_2)) \equiv W_{\otimes}[h(x_1), h(x_2)] \quad (3.4)$$

$$h_{x_1 \ominus 2} = f_{\ominus}(h(x_1), h(x_2)) \equiv W_{\ominus}[h(x_1), h(x_2)] \quad (3.5)$$

其中 $[h(x_1), h(x_2)]$ 表示拼接操作,  $W_{\oplus}$ 、 $W_{\otimes}$ 、 $W_{\ominus} \in \mathcal{R}^{K \times 2K}$ 分别表示“并”、“交”、“差”三个操作符的参数矩阵, 而 $h_{x_1 \oplus 2}$ 、 $h_{x_1 \otimes 2}$ 、 $h_{x_1 \ominus 2}$ 的标签分别为 $l_1 \oplus l_2$ 、 $l_1 \otimes l_2$ 、 $l_1 \ominus l_2$ 。

关键在于得到的新的哈希编码的类别

对于差来说

如果 $|l_1| > |l_1 \cap l_2|$ , 则等于 $l_1 - l_1 \cap l_2$ , 否则为 $l_1$

值得注意的是, “差”操作符相比于“并”“交”操作符有所不同, 因为当 $l_1 \cap l_2 = \emptyset$ 时 $l_1 - l_1 \cap l_2$ 没有意义, 因此对于差操作符以随机方式产生训练样本并不实用。为了解决这个问题, 我们在训练DCOH时将“差”操作符级联到“并”操作符之后, 如下所示:

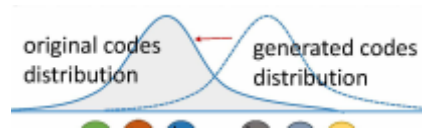
$$h_{x_1 \oplus 2 \ominus 2} = f_s(h_{x_1 \oplus 2}, h(x_2)) \equiv W_{\ominus}[h_{x_1 \oplus 2}, h(x_2)] \quad (3.6)$$

所以差运算放到并运算后面训练

三个损失函数

- 特别相似的图像的编码应该保持一致 (多层相似保持)
- 哈希码在语义空间应该是可区分的 (方便在语义层级上操作哈希码)
- 编码操作网络的需求, 输入和输出应该具有相同的分布 (输出在输入空间中检索最近邻样本)

图中表示原始编码与生成的编码的分布



传入到分类器D中, 是一个学习的损失函数, 对抗学习用于调整CON网络的参数