MABNET：用于图像检索的混合学习主网络辅助网络伙伴网络

Rohit Agarwal[†*] *, Gyanendra Das*[‡*]*, Saksham Aggarwal*[‡*]*, Alexander Horsch*[†]*, Dilip K. Prasad*[†]

[†]Bio-AI Lab，挪威北极大学 UiT，特罗姆瑟，挪威
[‡]印度理工学院（印度矿业学院），丹巴德，印度

arXiv:2303.03050v1 [cs.CV] 6 Mar 2023

摘要

近年来，图像检索受到越来越多的关注。现有方法分为监督式和自监督式。这些方法并未充分利用结合监督与自监督的混合学习优势。我们提出了一种用于图像检索的新型主助手伙伴网络（MABNet），融合了这两种学习机制。MABNet由主模块和助手模块组成，二者通过监督学习独立训练，并通过自监督学习协同训练。主网络将其知识库作为自监督学习的参考，以此指导辅助网络；辅助网络则通过权重迁移将其知识反馈给主网络。我们在公共数据集上进行了大量实验，涵盖使用和不使用后处理的情况。

索引词— 图像检索、监督、自监督、MABNet、视觉 Transformer

1. 引言

图像检索是指给定一幅无标签查询图像，从数据库中返回相关实例的任务。该任务既可采用监督方式，也可采用自监督方式。由于有真实标注，监督方法能够形成更优的决策边界，且通常基于残差网络等卷积神经网络（CNN） [1–12]，因为这类网络能够提取图像级描述符。

近年来，自监督学习已被用于解决图像检索任务 [13, 14]。DINO [13] 利用视觉 Transformer（ViT）的注意力机制进行自监督，使其知识模型包含有关图像语义分割的显式信息。

监督学习和自监督学习各有优势，而现有文献尚未探索在图像检索任务中同时运用这两种方法。自监督将通过明确的语义分割补充监督方法，从而改善决策边界。我们提出主网络辅助网络伙伴网络（MABNet），这是一种同时采用监督学习和自监督学习机制的新型模型，因此兼具这两种学习范式的优势

*共同贡献

。MABNet 是一种双模块伙伴网络（见图 1），其中一个是主网络，另一个是辅助网络。两者均通过监督学习独立学习。此外，它们使用距离度量比较各自的潜在特征以进行自监督，其中辅助网络以主网络的潜在特征作为自监督的参考。随后，辅助网络的知识模型通过权重迁移更新主模块的权重。从这个意义上说，辅助网络分担学习负荷，执行自监督，并以全面的知识模型协助主网络。此外，主网络和辅助网络各自专注于不同的学习内容：主网络使用全局和局部图像裁剪，而辅助网络专注于全局裁剪。

卷积是一种局部算子，而 Transformer 是一种全局算子，它会考虑所有像素，并通过注意力机制找出重要特征。众所周知，视觉 Transformer 的低层有效感受野远大于残差网络，这有助于其比卷积神经网络融入更多全局信息 [13, 14]。因此，我们在主网络和辅助模块中均采用视觉 Transformer。

一些方法采用多种后处理技术，例如平均查询扩展（AQE） [15], 、热扩散 [16], 、离线扩散 [17] 和 CVNet-Rerank [10]，通过生成相似图像的排序列表来获得最终结果。因此，我们将图像检索模型分为两类：不使用后处理技术的模型和使用后处理技术的模型。一些模型 [6–9] 也将后处理作为其端到端流程的一部分，因此我们将它们归入后一类。

贡献(1) 我们提出了 MABNet框架，该框架结合了监督学习和自监督学习范式。(2) 我们提出采用主网络和辅助模块：主网络专注于通用知识学习，而辅助网络则通过权重迁移帮助主网络完善其知识库。(3) 我们将所有图像检索模型分为采用后处理方法和不采用后处理方法两类。我们在 GLDv2-train-clean (GLDv2-TC) [18]数据集上训练 MABNet，并在常用基准数据集上验证其有效性，包括牛津（Ox5k 和 Ox105k） [19], 、巴黎（P6k 和 P106k） [20],、Revisited Oxford（ROx(M) 和 ROx(H)） [21] 以及重访巴黎

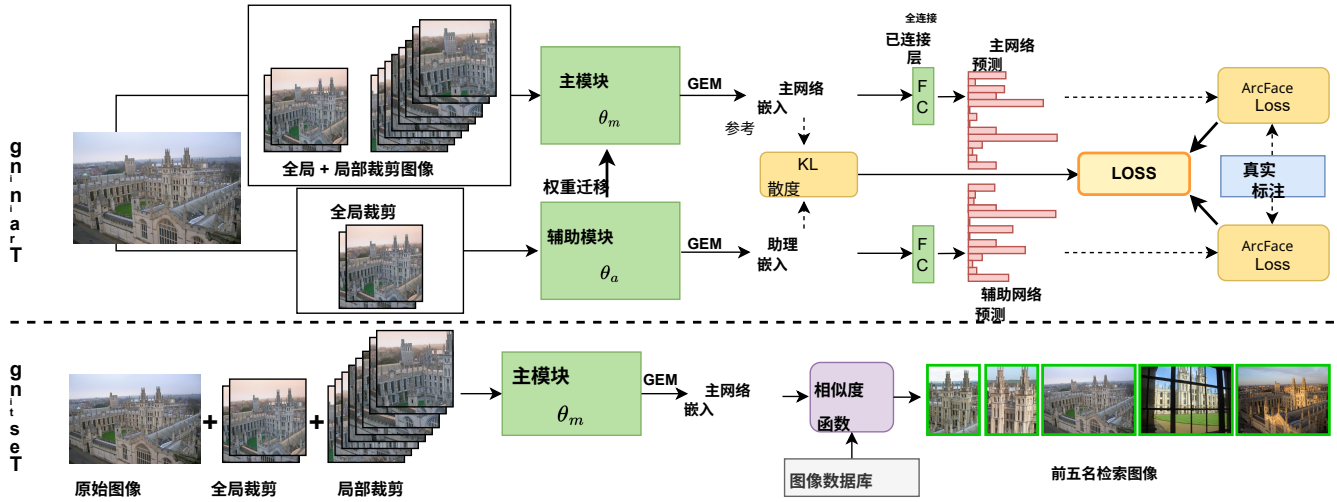

图1。训练架构（上部）：主助手伙伴网络（MABNet）的架构。全局裁剪图像输入辅助模块，而局部裁剪图像和全局裁剪图像均用于主模块，分别通过广义均值池化（GEM）层生成辅助嵌入和主嵌入。在这些嵌入上计算 KL 散度损失。随后，嵌入经过全连接层以获得预测，并在此基础上应用 ArcFace损失。每个训练周期结束后，通过权重迁移更新主模块的权重。测试架构（下部）：测试时仅通过主模块进行预测。将原始输入图像及其全局裁剪图像和局部裁剪图像输入训练好的主模块，从 GEM 层获取嵌入。随后，使用相似度函数（任意后处理方法）将这些嵌入的拼接结果与数据库中图像的嵌入进行比较，以检索 k 最佳图像。

（RPar(M) 和 RPar(H)） [21]这两类数据集。

2. 相关工作

基于CNN的表示是图像检索的一种颇具吸引力的解决方案。早期方法使用全连接（FC）层激活作为全局图像描述子。然而，全连接层无法学习几何不变性，并且会抑制空间信息。因此，可以认为全连接层之前的最后一个卷积层包含空间上下文。MAC [22] 通过选择最后一个卷积层并采用最大池化来引入几何不变性，从而提升了模型在图像检索任务上的性能。和池化卷积（SPoC） [23]使用特征图进行和池化。文献 [24]提出了包括跨维度加权和池化步骤在内的非参数加权方案。R-MAC [1] 采用混合方法，对各区域执行最大池化，并对区域描述子进行和池化。广义均值池化（GEM） [25], 广泛用于图像检索任务，通过池化参数将最大池化和平均池化进行了统一。多个全局描述符的组合（CGD） [26]通过拼接多个全局描述符（包括 SPoC、MAC 和 GEM）形成集成效应。因此，CGD实现了一种强大的图像检索混合方法。

在众多基于卷积神经网络的图像检索技术中，DELF [4] 和 DIR [3] 取得了最先进结果。DELF 是一种局部特征描述子，并结合了注意力机制，用于识别对图像检索具有语义价值的特征。另一方面，DIR 通过聚合许多区域级描述子，为每幅图像生成全局且紧凑的定长表示。它采用三流孪生网络 [27] ，通过三元组排序损失显式优化 R-MAC 表示的权重，以用于图像检索。它还学习 R-MAC 描述子的池化机制。选择性深度卷积特征 [5] 提出了使用各种掩码策略，从而选取具有代表性的局部卷积特征子集，并利用最先进的嵌入方法增强这些特征的判别能力。siaMAC [2] 是一种用于图像检索的卷积神经网络微调无监督方法，它利用三维重建来为卷积神经网络筛选训练数据。

可以将卷积特征图嵌入高维空间，以获得紧凑的特征，用于图像检索任务。常用的嵌入方法包括词袋（BoW） [28], 局部聚合描述符向量（VLAD） [29], 和 Fischer 向量 [30]。VLAD 捕获图像中聚合的局部描述符的统计信息。BoW 聚合会保留视觉词的计数，而 VLAD 则存储每个视觉词对应的残差之和（描述符与其对应聚类中心之间的差值）。NetVLAd [31]是一种广义 VLAD 层，其中包含可训练参数。

然而，卷积是一种局部算子，而 Transformer 是一种全局算子，它会考虑所有像素——

并通过应用注意力来寻找重要特征。研究 [32], 表明，视觉 Transformer 底层的有效感受野远大于残差网络，这有助于它们比卷积神经网络纳入更多全局信息。尽管视觉 Transformer 缺少卷积神经网络通过平移等变性和局部性所具备的归纳偏置，但它在分类、目标检测、目标跟踪和分割等各种任务上仍能取得有竞争力甚至更好的结果。最近，一些研究 [13, 14] 使用视觉 Transformer 进行有效的图像检索。DINO 的 [13] 知识模型包含图像语义分割的显式信息，因此性能更好。这些原因促使我们采用视觉 Transformer 作为模型的骨干网络。

一些研究也致力于设计专用于图像检索任务的损失函数。 [12]将二阶空间注意力与二阶描述符损失相结合，以改进用于检索和匹配的图像特征。 [33] 提出了一种优化目标，用于优化 Smooth-AP（平均精度的平滑近似），而非直接优化不可微的 AP。

为了提高图像搜索系统的准确率，人们也采用了许多后处理技术。查询扩展（QE） [34–36] 是实现这一目标的常用技术：将初始排序中得到的相关候选项聚合为扩展查询，从而为查询图像提供稳健的表示，再用它搜索数据库中的更多图像。聚合候选项可以强化它们之间的共享信息，并注入原始查询中没有的新信息。 [37]提出了另一种基于图神经网络的后处理技术。基于热扩散的图像重排序 [16]会对给定查询图像的若干个排名靠前的图像重新排序，并将查询视为热源，以避免重复特征被过度体现。扩散的一个缺点是，与朴素的 K 近邻搜索相比，其运行速度较慢。 [17] 通过预先计算数据库中每个元素的扩散结果解决了这一问题，使在线搜索只需在 K 近邻搜索的基础上进行简单的线性组合。该方法还提出采用延迟截断，即在归一化后进行截断，以获得更好的性能。

3. 方法

所提出的 MABNet（见图 1）由两个模块组成：主模块和辅助模块。这两个模块均以视觉 Transformer 为骨干网络（也可以采用残差网络等其他架构），记为 V，但分别使用不同的参数 θ_m 和 θ_a。主网络的作用是在关注局部空间上下文的同时保留全局上下文，以执行图像检索；辅助网络则通过权重共享提供全局空间信息，从而辅助主模块，并丰富所学习的表征

算法 1： MABNet 训练算法

输入： 预训练主模型块： V_{θ_m}，预训练辅助模块： V_{θ_a}，输入图像： x，真实标签： y，批量大小： N，权重迁移参数： λ

初始化： KL 散度损失： $L_{KL}=0$，辅助网络 ArcFace损失： $L_{AF}^{A}=0$, 主模型ArcFace损失: $L_{AF}^{M}=0$, 总损失： $Loss=0$

对于 $n=1$**到** N **执行**

- C_G =多裁剪$_G$(x_n)　// 全局裁剪
- C_L= 多裁剪$_L$(x_n)　// 局部裁剪
- $L_{KL}=0$、$L_{AF}^{M}=0$、 $L_{AF}^{A}=0$
- **对于** $i=1$ **到** $|C_G|$，**执行**
  - E_a^i = V_{θ_a}(x_a^i)； $x_a^i \in \{C_G\}$
  - // 助理嵌入
  - y_a^i =全连接层$_a$(E_a^i)
  - L_{AF}^{A} += ArcFace 损失(y_a^i, y_n)
- **end**
- **对于** $j=1$ **到** $|C_G \cup C_L|$ **执行**
  - E_m^j = V_{θ_m}(x_m^j)； $x_m^j \in \{C_G \cup C_L\}$
  - // 主嵌入
  - y_m^j =全连接层$_m$(E_m^j)
  - L_{AF}^{M} += ArcFace 损失(y_m^j, y_n)
- **end**
- $i \in \{1,...,|C_G|\}$ **用于**以及 $j \in \{1,...,|C_G \cup C_L|\}$ **do**
  - L_{KL} += KL 散度损失(E_a^i, E_m^j)
- **end**
- $Loss+=L_{KL}+L_{AF}^{M}+L_{AF}^{A}$

end

$\frac{\partial Loss}{\partial \theta_m}$, $\frac{\partial Loss}{\partial \theta_a}$　// 反向传播更新参数 θ_m 和 θ_a

$\theta_m = \lambda * \theta_m + (1-\lambda)\theta_a$ // 权重迁移

主网络所学习的表征。

在 MABNet 中，我们不再将原始输入图像 x直接传入网络，而是使用多裁剪方法对输入图像进行多次裁剪。我们将覆盖原始图像不足 50% 的裁剪图像称为局部裁剪（记为 C_L），其余裁剪图像均称为全局裁剪（C_G）。主网络接收所有裁剪图像，而辅助网络只处理全局裁剪。对于输入图像 x，我们从分别对应主网络和辅助模块的 GEM层 [25]中得到两个嵌入 E_m 和 E_a。 E_m 和 E_a 由以下公式给出，其中$x_m \in \{C_G \cup C_L\}$ 和 $x_a \in C_G$。

$$E_m(x) = V_{\theta_m}(x_m) \quad ; \quad E_a(x) = V_{\theta_a}(x_a), \tag{1}$$

Kullback–Leibler（KL）散度损失 [38]是在嵌入 E_m 和 E_a之间计算的。一个全连接线性层将嵌入转换为相应的输出标签。我们采用 ArcFace [39] 来获得监督

算法 2： MABNet 测试算法

输入： 充分训练的主模块：V_{θ_m}，图像位于数据库：X，查询图像：q，要检索的图像数量：k

输出： 与查询图像相似的前 k 张图像

对于 $x_i \in X$，**执行**

  $E_i = V_{\theta_m}(x_i)$　　// 创建数据库嵌入

end

E $= [E_1, E_2, ... E_{|X|}]$

$E^q = V_{\theta_m}(q)$　　// 查询嵌入向量

输出 = 相似度函数(E^q, E, k)

其中，SimilarityFunc 是后处理方法如第 3.4 节所述

损失。每个模块相对于 ArcFace损失的梯度独立传播，而相对于 KL 损失的梯度则共同传播。此外，我们采用从辅助网络到主网络的权重共享技术，以更新主网络模块的知识模型。最终预测由主网络输出，因为与辅助网络相比，主网络捕获的信息更具泛化性。MABNet 的架构如图 1 所示，训练和测试的伪代码见算法 1 和 2。

3.1. 图像裁剪

我们从原始图像中生成局部裁剪和全局裁剪，并将图像裁剪的不同组合传递给各个模块。使用裁剪图像而非完整输入图像可以扩充数据。拍摄同一物体的图像时，角度可能各不相同，因此这些图像之间的相似度较低。多裁剪策略 [40] 通过比较图像各部分的嵌入，增加了识别这些图像的机会。此外，局部裁剪为主网络模块提供上下文信息，而辅助模块则包含丰富的全局裁剪空间信息。

3.2. 我们模型训练中使用的损失函数

我们在此讨论 MABNet 中使用的两种损失，即 KL散度损失 [38] 和 ArcFace损失 [39]。

KL 散度损失 由于两个模块是在不同的图像裁剪组合上训练的，因此它们的嵌入表示了不同的特征。辅助模块包含空间信息，而主模块的嵌入则具有更强的泛化性。我们在两个模块的嵌入上引入了自监督 KL 散度损失。主模块的嵌入 E_m 被用作参考分布，因为它包含更具泛化性的信息。KL 散度计算相对熵，用以表明嵌入 E_a 与 E_m之间的差异。KL 散度损失为

输入图像 x 定义为

$$L_{KL}(x) = \sum_{x_m \in \{C_G \cup C_L\}} \sum_{x_a \in C_G} E_m(x_m) log \frac{E_m(x_m)}{E_a(x_a)} \quad (2)$$

其中符号定义同前。通过 KL 散度损失，我们引入了局部到全局对应关系，这有助于辅助网络学习更精细的信息。随后，这些信息通过权重迁移方法传回主模块。因此，主网络和辅助网络的学习相辅相成。

ArcFace损失 我们在两个模块中均采用有监督 ArcFace损失，以获得适用于图像检索任务的更丰富嵌入。softmax 损失不会显式地约束类内相似性和类间多样性。这会降低检索与识别任务的性能，尤其是在类内差异较大的情况下（如 [41], [42], [43]）。相比之下，ArcFace 通过增大类别之间的角度间隔来学习判别性特征。此外，由于其与嵌入超球面上的测地距离精确对应，因此具有清晰的几何解释。ArcFace 会变换 logits，将权重和嵌入特征归一化至单位模长，并将嵌入特征重新缩放至值 s。此外，在嵌入特征 y_a^i 与第 y_i 类的 ArcFace 头部权重 W_{y_i} 之间加入加性角度间隔惩罚 m，以同时增强类内紧凑性和类间差异。在我们的框架中，这些超参数 s 和 m均可学习。主模块中输入图像 x 的 ArcFace 损失定义为

$$L_{AF}^M(x) = -\frac{1}{|C|} \sum_{i=1}^{|C|} log \frac{e^{s(cos(\theta_{y_i}+m))}}{e^{s(cos(\theta_{y_i}+m))} + \sum_{j=1, j\neq y_i}^{n} e^{scos(\theta_j)}} \quad (3)$$

其中，$C = \{C_G \cup C_L\}$ 是传入主模块的裁剪图像集合，$|\cdot|$ 表示其基数，n 是类别标签的数量，$W_j^T x_i$ 是每个类别的 logit $cos(\theta_j)$，而θ_{y_i} 是 y_a^i 与真实权重 W_{y_i} 之间的夹角。类似地，辅助模块的 ArcFace损失 $L_{AF}^A(x)$ 遵循上述定义，但使用裁剪图像集合$C = C_G$。

总损失 批量大小为 N 时，模型的总损失为

$$L = \sum_{k=1}^{N} \{L_{KL}(x_k) + L_{AF}^M(x_k) + L_{AF}^A(x_k)\} \quad (4)$$

3.3. 权重迁移

两个模块均通过反向传播，根据两种损失学习权重：自监督 KL 散度损失和有监督 ArcFace损失。除了通过梯度下降进行更新外，

主模块的权重还通过主网络和辅助网络参数的加权平均进行更新，具体如下：

$$\theta_m = \lambda\theta_m + (1-\lambda)\theta_a \tag{5}$$

其中，λ 是权重迁移参数。λ 的值由网格法确定，范围为 $0 \leq \lambda \leq 1$。

3.4. 测试阶段

在推理时，我们将输入图像及其所有裁剪图像（即 C_G 和 C_L ）输入主模块进行前向传播，以获取来自GEM层的嵌入。因此，我们得到C_G+C_L+1 个嵌入。我们将它们全部拼接起来。每张图像的嵌入大小为512。然后，我们在这些嵌入上训练一个简单的K近邻模型，并找出与输入图像最近的邻居。或者，我们也可以使用任何后处理技术，通过生成数据集中相似图像的排序列表来获得最终结果。图像检索任务中最广泛使用的后处理技术包括平均查询扩展（AQE） [15], 热扩散 [16],离线扩散 [17] 和CVNet-Rerank [10]。

3.5. 模型工作原理

图2展示了主模块和助手模块渐进式学习的示例。由于主嵌入最初同时接收全局裁剪和局部裁剪，因此其表示更加通用且分布更广。助手嵌入的分布也较广，但仅覆盖图像的一部分，具体取决于传递给它的全局裁剪。不过，助手模块会通过自监督将其记录的信息与主模块进行比较，因此能够学习局部信息。从后期训练轮次的热力图可以看出，随着学习进行，助手模块会越来越聚焦于某个特定区域。反过来，主模块通过权重迁移从助手模块获取这些局部信息，因此在后期训练轮次中也会更加聚焦，从而形成更好的决策边界。图1下方的面板和图2(b)展示了从数据库中检索到的图像示例。此外，还展示了检索失败时的嵌入，以供参考。

4. 数据集

我们在 Google地标数据集v2（GLDv2）的 train-clean 版本上训练模型（预训练于 ImageNet），并在牛津、巴黎、Revisited Oxford、重访巴黎和 GLDv2 数据集上进行测试。

牛津建筑数据集 我们使用 Oxford5k 和 Oxford105k 两个数据集。Oxford5k 包含 5,062 张牛津建筑图像 [44]。这些图像采集自 Flickr。该数据集有 55 张查询图像和 5 个查询，每个查询都对应一个地标及其相应的真实标注。Oxford105k 是 Oxford5k 数据集的扩展，

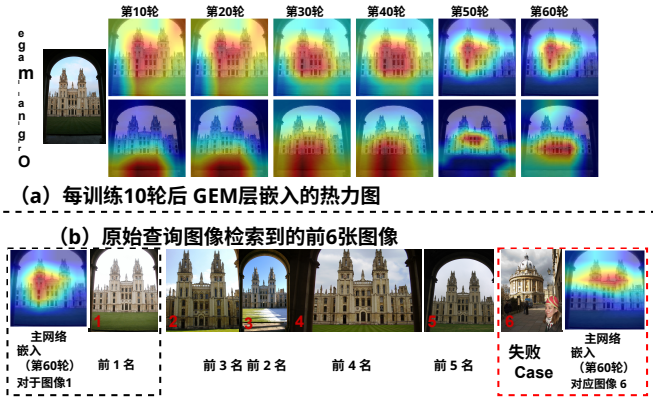

图 2。（a）使用 Grad-CAM，分别从主嵌入和助手嵌入生成原始图像每训练 10 个训练轮次后的叠加热力图，直至 60 个训练轮次。（b）原始查询图像检索到的前 5 张图像。我们还展示了排名第 6 的检索图像，这是一个失败案例。左侧和右侧热力图分别来自排名第 1 的检索图像和失败案例的主嵌入。该结果是在未经后处理的 Ox105k 数据集上获得的。

其中还额外包含 99,782 张从 Flickr 获取的负样本图像 [45]。

巴黎数据集 我们报告了巴黎数据集 [45] 中 Paris6k 和 Paris106k 上的结果。Paris6k 包含 6,392 张巴黎市的高分辨率图像。与 Oxford5k 类似，该数据集通过查询与著名巴黎地标相关联的文本标签，从 Flickr 收集而来。该数据集还提供了 55 张查询图像及其对应的真实标注图像。Paris106k 图像是 Paris6k 数据集的扩展，包含额外的 100,000 张 Flickr 图像。

重访牛津和巴黎数据集 [21] 牛津和巴黎数据集近期经过修订，修正了标注错误、调整了数据集规模并提升了挑战难度。重访数据集 ROxford 和 RParis 分别包含 4,993 张和 6,322 张图像，并各自采用不同的查询集，每个查询集均包含 70 张图像。评估协议分为简单、中等和困难三个难度级别。我们报告中等和困难这两个具有挑战性的类别中的结果。

Google地标数据集v2 我们使用 Google地标数据集v2（GLDv2）的 train-clean 版本 [18]，该版本在 Kaggle 挑战赛*“Google地标检索 2021年”*中发布。GLDv2 是一个大规模数据集，包含超过 500 万张地标图像，可用于实例级识别和检索任务。GLDv2-train-clean [46]是 GLDv2 的一个规模更小、更干净且经过进一步完善的版本，它是在采用 [47]提出的某些预处理步骤后构建的，旨在使每个类别在视觉上更加一致。GLDv2-train-clean 数据集包含 1,580,470 张训练图像、117,577 张测试图像和 81,313 个标签。

5. 实验

5.1. 对比方法

我们在两种情形下比较我们的方法：基础形式（不使用任何后处理方法）以及对结果进行后处理的形式。在基础形式下，我们将其与最先进的方法进行比较，例如 DIR [3], 、DELF [4], 、Deep Conv [5], 、siaMAC [2], 、R-MAC [1], 、Listwise Loss [11], 、IRT(R) [14], 、DINO [13]和 SOLAR [12]。随后，我们通过应用后处理方法来比较我们的模型。此处的比较对象包括 DELG [6], 、DOLG [7], 、Swin-T-DALG [8], 、Swin-S-DALG [8], 、GeM [9] 和 CVNet-Rerank [10] ，这些方法均将后处理纳入其处理流程。

5.2. 设置

我们采用 2 个全局裁剪图像和 8 个不同大小的局部裁剪图像。对于全局裁剪图像和局部裁剪图像，随机缩放的范围分别为 [0.5, 1] 和 [0.05、0.5]。全局裁剪图像和局部裁剪图像分别调整为 224×224 和 96×96。我们使用在 ImageNet 上预训练的 ViT-small 作为骨干网络。输入图块大小为 16×16 的图像块会被展平，并投影为 M 个低维线性向量。在将其输入 Transformer编码器之前，通过向线性嵌入中添加可训练的位置编码来提供位置先验。模型使用 $8\times$ 个 V100 训练 100 个训练轮次，批量大小为 256。我们使用 AdamW优化器，学习率为 0.0005，权重衰减为 10^{-5}。此外，前 10 个训练轮次采用线性预热，之后学习率遵循余弦调度。我们采用混合精度训练以加快计算速度。λ 设置为 0.5。我们采用标准的平均精度均值（mAP）指标来报告准确率。在 GLDv2-TC [18] 上的训练时间为 2.5 天。浮点运算次数为 9.2 GFlops。单张图像所有拼接嵌入的推理时间为 132 ms，而仅使用原始图像时为 11 ms。

5.3. 无后处理

牛津和巴黎数据集表1上半部分展示了我们在牛津和巴黎数据上的结果。我们的模型显著超越了已报告的最先进性能（在 Ox105k 和 P106k 数据集上分别高出 6.1% 和 7.3%）。我们的模型在 Ox5k 和 P6k 上的 mAP 分数也最高。重访数据集MABNet 的表现优于其他模型，在 ROx(M) 上高出 15.1%，在 ROx(H) 上高出 13.3%，在 RPar(M) 上高出 3 %，在 RPar(H) 上高出 8.4%（见表1）。

5.4. 使用后处理

牛津和巴黎数据集表2中的青色行显示，后处理技术提升了 MABNet 的性能。离线扩散带来了最佳提升

表1。在牛津、巴黎及其重访数据集上，各种方法未经后处理时的 mAP 分数。MABNet-R50 表示以残差网络-50 为骨干网络的 MABNet。

模型	Ox5k	Ox105k	P6k	P106k
R-MAC [1]	66.9	61.6	83.0	75.7
siaMAC [2]	80.0	75.1	82.9	75.3
DIR [3]	83.1	78.6	87.1	79.7
DELF [4]	83.8	82.6	85.0	81.7
Deep Conv [5]	83.8	80.6	88.3	83.1
MABNet	**91.5**	**88.7**	**94.2**	**90.4**
	ROx(M)	ROx(H)	RPar(M)	RPar(H)
DINO [13]	51.5	24.3	75.3	51.6
IRT(R) [14]	55.1	28.3	72.7	49.6
Listwise [11]	67.5	42.8	80.1	60.5
SOLAR [12]	69.9	47.9	81.6	64.5
MABNet	**85.0**	**61.2**	**84.6**	**72.9**
MABNet-R50	80.3	59.2	82.0	67.4

(3.9−7.6%)。在采用相同后处理方法时，MABNet 优于其他方法。重访数据集表2下半部分提到的所有方法，要么在推理后进行后处理，要么将后处理技术纳入模型的一部分。因此，我们将它们归入后处理类别。从表2可以看出，在大多数情况下，采用后处理的 MABNet 表现优于其他方法。

5.5. GLDv2数据集上的结果

我们还将模型应用于 GLDv2数据集 的测试数据，并比较了有无后处理的情况。由于这是一个相对较新的数据集，可供比较的模型不多。Zhang等人 [48] 报告的mAP分数为32.4%，而MABNet的mAP分数为37%（见表3）。后处理方法，尤其是离线扩散，使模型性能进一步提升了1.1%。

6. 消融研究

我们在Ox5K数据集上对MABNet进行了多种消融研究（除非另有说明，均未进行后处理）。

6.1. 权重迁移

我们考虑了不将权重迁移从辅助网络传递至主网络的训练方式。得分从 91.5% 降至 81.9%，这表明了其重要性。我们还研究了改变权重迁移方向的影响，即通过权重迁移更新辅助网络。结果得分降至 80.5%。我们推测，相较于辅助网络，迁移主网络的权重更为有效，因为

表2。不同方法采用不同后处理技术时，在牛津、巴黎及其重访数据集上的mAP分数。图例：PP－后处理，Q－AQE，AQ－AML+ AQE，DQ－DIR +AQE，WQR－HeW + AQE + HeR，O－离线扩散，CV-CVNet。

模型 + PP	Ox5k	Ox105k	P6k	P106k
MABNet	91.5	88.7	94.2	90.4
siaMAC + Q [2]	85.4	82.3	87.0	79.6
DIR +Q [3]	89.0	87.8	93.8	90.5
MABNet + Q	92.3	89.6	95.2	91.1
R-MAC + AQ [1]	77.3	73.2	86.5	79.8
DELF + DQ [4]	90.0	88.5	95.7	92.8
siaMAC+WQR [16]	92.0	90.3	94.3	90.2
MABNet + WQR	95.7	92.9	96.5	95.6
R-MAC + O [17]	96.2	95.2	97.8	96.2
MABNet + O	**97.2**	**96.3**	**98.1**	**96.8**
	ROx(M)	ROx(H)	RPar(M)	RPar(H)
DELG [6]	81.2	64.0	87.2	72.8
DOLG [7]	81.5	61.1	91.0	80.3
Swin-T-DALG [8]	78.7	54.7	88.2	76.3
Swin-S-DALG [8]	79.9	57.5	90.4	79.0
GeM（基线） [9]	83.0	65.5	90.2	80.7
GeM-局部匹配 [9]	85.9	71.2	92.0	**83.7**
CVNet-Rerank [10]	87.2	75.9	91.2	81.1
MABNet + Q	87.1	63.8	86.4	73.5
MABNet + WQR	88.4	65.4	88.3	75.8
MABNet + O	**89.3**	66.2	88.9	78.2
MABNet + 计算机视觉	87.1	**76.5**	**92.3**	82.7

将来自辅助网络的仅包含全局的空间信息添加到主网络的局部—全局上下文嵌入中，可以将其丰富为一个更具泛化能力的模块。

6.2. KL散度损失

我们研究了训练模型时去掉 KL 损失的影响。性能从 91.3% 降至 83.3%，表明自监督是决定 MABNet 性能的关键因素。我们还研究了反转 KL 损失的参考对象，即将助手嵌入设为参考。该模型的表现比 MABNet 差 5.8%，这表明以主网络为参考非常重要。

6.3. 辅助模块

我们仅使用主模块训练方法，移除辅助模块及其所有依赖连接，例如权重迁移、KL 和 ArcFace损失。为公平比较，我们纳入所有裁剪图像和原始图像。这里的分数有所下降：Ox5k 从 91.5% 降至 80.2%，ROx(M) 从 85.0% 降至 77.6%，ROx(H) 从 61.2% 降至 49.3%，这表明辅助模块至关重要。

表3。MABNet 经与未经后处理的平均精度均值，以及其他模型在 GLDv2数据集测试图像上的表现。

模型	后处理	GLDv2
残差网络101+ArcFace [18]		28.5
Zhang等人 [48]		32.4
DELG [6]		26.8
MABNet		**37.0**
MABNet	AQE	37.2
MABNet	HeW+AQE+HeR	37.8
MABNet	离线扩散	38.1

表4。MABNet(C) 在牛津、巴黎及其重访数据集上未进行后处理时的平均精度均值。

模型	Ox5k	Ox105k	P6k	P106k
MABNet(C)	92.5	89.6	95.3	92.1
MABNet(C) + Q	93.6	92.4	95.9	94.1
MABNet(C) + WQR	97.1	93.8	98.1	96.4
MABNet(C) + O	98.3	97.8	98.3	97.5
	ROx(M)	ROx(H)	RPar(M)	RPar(H)
MABNet(C)	85.8	62.1	85.7	73.8
MABNet(C) + Q	88.1	62.8	86.2	74.7
MABNet(C) + WQR	89.7	65.8	87.6	76.9
MABNet(C) + O	90.2	67.5	89.8	79.4

6.4. 视觉 Transformer 骨干网络

鉴于视觉 Transformer 的优势（见第 1 节），我们将其用作骨干网络。我们以残差网络-50 替代视觉 Transformer，并采用相同的训练流程。表1显示，视觉 Transformer 的性能显著优于残差网络-50。尽管如此，采用残差网络-50 的 MABNet 仍优于其他方法。

6.5. 拼接特征

推理时，我们使用原始图像、2 个全局裁剪和 8 个局部裁剪。MABNet 在 Ox5k 上的 mAP 分数为 91.5%。仅将原始图像作为推理输入时，mAP 分数为 90.2%，仍优于最先进水平。仅输入 2 个全局裁剪时，mAP 分数为 85.7%；仅输入 8 个局部裁剪时，则为 85.4%。将原始图像与 4 个全局裁剪和 12 个局部裁剪的特征进行拼接后，mAP 分数达到 91.7%，相比 91.5% 并无显著提升。我们由此认为，使用原始图像、2 个全局裁剪和 8 个局部裁剪，能更好地平衡复杂度与性能。若要缩短推理时间，也可以只使用原始图像，并获得令人满意的结果。

表 5。不同 λ 值下的 mAP分数。此处报告的所有结果均未采用任何预处理技术。

λ	Ox5k	ROx(M)	ROx(H)	RPar(M)	RPar(H)
0.30	75.9	79.9	57.6	80.4	68.5
0.35	81.8	81.5	57.3	81.6	71.2
0.40	87.4	84.3	59.4	84.1	70.8
0.45	90.2	85.8	60.3	83.2	72.5
0.50	91.5	85	61.2	84.6	72.9
0.55	92.7	**87.3**	64.5	85.4	**75.2**
0.60	**93.2**	86.4	**67.8**	86.7	74.6
0.65	92.9	85.8	66.2	**87.3**	73.7
0.70	92.0	83.4	60.6	85.2	72.5
0.75	90.6	82.2	59.2	84.5	71.3
0.80	88.1	76.5	58.4	83.2	70.8
0.85	79.6	75.3	57.6	81.4	68.5
0.90	78.5	72.1	55.9	80.9	66.7
0.95	78.2	70.2	53.2	79.5	65.2

6.6. 两个模块的集成

MABNet 仅使用主模块生成最终输出。这里，我们将 MABNet 扩展为 MABNet(C)：通过拼接主模块和辅助模块的嵌入来进行预测。可以看到，MABNet(C) 在所有情况下表现都更好（见表 4）。推理时，MABNet 的速度是 MABNet(C) 的两倍，因为其参数数量只有后者的一半。选择 MABNet 还是 MABNet(C)，取决于推理时间与准确率提升之间的权衡。

6.7. λ 对 MABNet 的影响

λ 的值决定了权重迁移对 MABNet 的影响。如果 λ 为 1，则权重迁移对模型没有影响，因为没有信息从辅助网络迁移到主网络。而当该值为 0 时，主模块的参数会被辅助模块替换，因此每个 epoch 结束后，我们得到的 ViT 模块完全相同。

在我们的主要实验中，我们使用 $\lambda = 0.5$ 的值，因此权重迁移是主网络参数和辅助参数的平均值。在此基础上，我们进一步针对 λ 的不同取值优化方法。我们在 Oxford5k 数据集以及重访牛津和巴黎数据集上报告了结果，并表明经过充分优化后，MABNet 在所有数据集上的结果都能进一步提升。在这里，我们将 λ 的取值从 0.3 变化到 0.95，步长为 0.05。Oxford5k 数据集上不同 λ 取值的结果如表 5 所示。可以看出，与平均值相比，在 $\lambda = 0.6$ 处取得了最佳结果（提升了 1.7%）。mAP 分数从 $\lambda = 0.3$（75.9%）持续提升至 $\lambda = 0.6$（93.2%），之后便持续下降。当 λ 达到最大值，即 $\lambda = 0.95$ 时，我们得到 78.2% 的 mAP 分数。A

在重访数据集上也可以观察到类似的分数趋势：当 λ 处于 0.55 到 0.65 之间时，取得最高分数。结果如表 5 所示。因此，通过进一步优化 λ 的取值，可以进一步提升 MABNet 的性能。

6.8. MABNet 的不同特征组合

根据不同特征，我们的模型存在许多可能的设置组合：(a) 输入主网络和辅助模块的裁剪图像类型（全局裁剪、局部裁剪或两者兼有）；(b) 是否存在权重迁移机制；(c) 若存在权重迁移，其迁移方向；(d) 主网络与辅助模块之间是否存在自监督损失（即 KL 散度损失）；(e) KL 散度损失中参考分布的选择。

在我们的设置中，我们向主模块输入全局裁剪和局部裁剪，而仅向辅助模块输入全局裁剪。我们假设，在使用 KL 散度损失时，将主模块的嵌入分布作为参考，有助于辅助模块学习更详细的信息。由于从辅助模块到主模块的权重迁移机制，这进一步有助于主模块获得更强的泛化能力。

为了验证我们的假设，我们进行了一系列实验，探索上述讨论的所有不同类型的组合。鉴于主模块产生最终输出，主模块必须从辅助模块中获得某种学习。因此，我们仅考虑那些至少包含一种从辅助网络到主网络的学习支持的组合：要么根据辅助模块更新主模块的参数，要么 KL 损失的方向指向主模块，即将助手嵌入视为参考分布。所有组合的结果如表 6 所示。在 Oxford5k 数据集上，使用 $\lambda = 0.5$，仅通过主模块进行推理。以青色标示的 MABNet 的表现显著优于其他所有设置，从而验证了我们的假设。

6.8.1. 裁剪图像的类型

将全局裁剪和局部裁剪图像输入主模块，并将全局裁剪图像输入辅助模块，与其他所有模块相比，能取得最佳结果；但在某些设置下，仅将全局裁剪图像输入主模块、并将所有裁剪图像输入辅助模块的情况除外。即使比较这种情况，MABNet 的表现仍然最佳，这证明了为主模块和辅助模块选择这些裁剪图像是合理的。

6.8.2. 权重迁移与 *KL* 散度损失

在此，我们分析权重迁移和 KL 散度损失的有无及其方向的不同组合对模型性能的影响。

反转 KL 散度方向并取消权重迁移 我们移除了权重迁移，并反转 KL 散度损失的方向，以检验权重迁移机制的重要性。尽管在没有权重迁移的情况下，KL 散度损失确实有助于辅助网络将知识迁移给主模块，但在几乎所有情况下，这仍会导致最差的性能。这表明权重迁移是我们网络中最重要的组成部分之一。

反转信息流在 MABNet 的原始设置中，我们使用主网络和辅助网络权重的加权平均来更新主模块，并以主模块的嵌入为参考，计算主模块与辅助模块嵌入之间的 KL 散度损失。在这项消融研究中，我们通过反转这一信息流来研究其对模型性能的影响：即将权重的平均值用于更新辅助模块，并以辅助模块的嵌入作为 KL 散度损失的参考。我们发现，得分降至 80.5%。我们推测，MABNet 中的信息流之所以更有效，是因为我们通过权重迁移，将辅助网络仅包含全局信息的空间信息融入主模块的局部—全局上下文嵌入，使其具有更强的通用性。此外，由于主网络能够获得通用嵌入，因此它通过将自身嵌入作为参考，利用 KL 散度来教导辅助模块。

7. 结论

我们提出了 MABNet，这是一种同时采用监督学习和自监督学习范式的新方法。此外，与以往提出的教师-学生模型相比，我们在伙伴模型中引入了机制，以有指导的方式进行协同学习。大量消融研究清楚地表明，伙伴模型的作用、主网络对自监督学习的引导，以及从辅助网络到主网络的知识传播都至关重要。

我们展示了与以往方法相比的显著性能提升，并在牛津和巴黎数据集及其重新评测版本上，取得了超越最先进水平的平均精度均值。我们还表明，后处理能够进一步提升性能。此外，我们在 Google Landmark 数据集上创下新的最先进水平：不进行后处理时达到 37%，使用离线扩散进行后处理时达到 ∼38%。

本质上，这项工作提出了一种基于视觉 Transformer 的强大学习范式。我们期待所提出的新颖范式能够推动基于视觉 Transformer 的架构进一步发展。

我们展示了如何通过多视图训练让两个视觉 Transformer 高效利用从 CNN 骨干网络中提取的局部信息，并强调了在面对长尾类别分布和类内变异性时，子中心 ArcFace 的优越性。我们确认了其适用性

表 6。此处展示了 MABNet 特征的不同排列与组合所取得的结果。本文使用缩写表示部分变量：KL 表示 Kullback-Leibler 损失，WT 表示权重迁移，mAP 表示模型的平均精度均值，主作物表示输入主模块的裁剪图像，助手模块裁剪图像表示输入辅助网络的裁剪图像。表中符号的含义如下：G 表示全局裁剪，L 表示局部裁剪，X 表示模型中不存在该特征，↑arrow 表示特征方向向上，即如果 KL 向上，则以助手嵌入作为参考；↓ arrow 表示方向向下，即如果 WT 向下，则辅助模块的参数会更新为加权平均值。与其他设置相比，MABNet（以青色标示）的表现最佳。此处结果基于 Oxford5k 数据集，且未采用任何后处理技术。

主作物	辅助网络裁剪图像	KL	WT	mAP
G+L	G	X	↑	83.3
		↓	↑	**91.5**
		↑	X	64.1
		↑	↓	80.5
		↑	↑	85.7
G+L	L	X	↑	61.9
		↓	↑	64.2
		↑	X	31.7
		↑	↓	61.6
		↑	↑	57.8
G	G+L	X	↑	69.2
		↓	↑	73.3
		↑	X	69.7
		↑	↓	84.2
		↑	↑	76.1
G	L	X	↑	52.5
		↓	↑	49.3
		↑	X	31.9
		↑	↓	31.4
		↑	↑	42.6
L	G+L	X	↑	55.4
		↓	↑	52.7
		↑	X	30.5
		↑	↓	48.7
		↑	↑	34.6
L	G	X	↑	39.9
		↓	↑	29.6
		↑	X	25.2
		↑	↓	23.4
		↑	↑	30.1

重新排序和扩散后处理技术带来了性能提升。消融研究证实，权重更新的方向以及嵌入之间的熵损失

对于架构至关重要。此外，使用拼接嵌入也取得了较高的指标。代码可在**https://github.com/Rohit102497/MABNet**获取。

致谢 我们感谢以下资助：科学振兴研究员项目资助（编号：325741，Dilip K. Prasad）、UiT主题项目 VirtualStain（Cristin ID：2061348，Alexander Horsch、Dilip K. Prasad），以及 Horizon 2020 FET开放资助项目 OrganVision（编号：964800）。

8. 参考文献

[1] Giorgos Tolias、Ronan Sicre 和 Herv´e J´egou，《通过 CNN 激活的积分最大池化进行特定物体检索》，载于国际学习表示会议，2016年。

[2] Filip Radenovi´c、Giorgos Tolias 和 Ond˘rej Chum，《CNN图像检索从词袋模型中学习：利用困难样本进行无监督微调》，载于欧洲计算机视觉会议。施普林格，2016年，第3–20页。

[3] Albert Gordo、Jon Almaz´an、Jerome Revaud 和 Diane Larlus，《深度图像检索：学习用于图像搜索的全局表征》，载于欧洲计算机视觉会议，2016年，第241–257页。

[4]Hyeonwoo Noh、Andre Araujo、Jack Sim、Tobias Weyand 和 Bohyung Han，《利用注意力深度局部特征进行大规模图像检索》，载于*IEEE*计算机视觉国际会议，2017年，第 3456–3465 页。

[5] Tuan Hoang、Thanh-Toan Do、Dang-Khoa Le Tan 和 Ngai-Man Cheung，《用于图像检索的选择性深度卷积特征》，载于*ACM*多媒体国际会议，2017年，第 1600–1608 页。

[6] Bingyi Cao、Andre Araujo 和 Jack Sim，《统一用于图像搜索的深度局部与全局特征》，载于欧洲计算机视觉会议。施普林格，2020年，第 726–743 页。

[7] Min Yang、Dongliang He、Miao Fan、Baorong Shi、Xue-tong Xue、Fu Li、Errui Ding 和 Jizhou Huang，《DOLG：通过局部与全局特征的深度正交融合实现单阶段图像检索》，载于*IEEE/CVF*国际计算机视觉会议论文集，2021年，第 11772–11781 页。

[8] Yuxin Song、Ruolin Zhu、Min Yang 和 Dongliang He，“DALG：用于图像检索的深度注意力局部与全局建模”，*arXiv* 预印本 *arXiv:2207.00287*，2022年。

[9] Zechao Hu 和 Adrian G Bors，“用于图像搜索的表达性局部特征匹配”，载于*2022*年第*26*届国际模式识别会议（*ICPR*）。IEEE，2022年，页码 1386–1392。

[10] Seongwon Lee、Hongje Seong、Suhyeon Lee 和 Eun-tai Kim，“用于图像检索的相关性验证”，载于*IEEE/CVF*计算机视觉与模式识别会议论文集（*CVPR*），2022年6月，页码 5374–5384。

[11] Jerome Revaud、Jon Almaz´an、Rafael S Rezende 和 Cesar Roberto de Souza，“ 通过平均精度进行学习：使用列表式损失训练图像检索模型”，载于*IEEE/CVF*国际计算机视觉会议论文集，2019年，第5107–5116页。

[12] Tony Ng、Vassileios Balntas、Yurun Tian 和 Krystian Mikolajczyk，“用于图像检索的 SOLAR：二阶损失与注意力”，载于欧洲计算机视觉会议。施普林格，2020年，第253–270页。

[13] Mathilde Caron、Hugo Touvron、Ishan Misra、Herv´e J´egou、Julien Mairal、Piotr Bojanowski 和 Armand Joulin，“自监督视觉变换器的新兴特性”，载于*IEEE*国际计算机视觉会议，2021年，第9650–9660页。

[14] Alaaeldin El-Nouby、NataliaNeverova、Ivan Laptev 和 Herv´e J´egou，《训练视觉 Transformer 进行图像检索》，2021年。

[15] Ond˘rej Chum、Andrej Mikulik、Michal Perdoch 和 Ji˘r´ı Matas，《Total recall II：重新审视查询扩展》，载于 *CVPR 2011*。IEEE，2011年，第889–896页。

[16] Shanmin Pang、Jin Ma、Jianru Xue、Jihua Zhu 和 Vi- centeOrdonez，《利用热扩散进行深度特征聚合和图像重排序以实现图像检索》，*IEEE*多媒体汇刊，第21卷，第6期，第1513–1523页，2018年。

[17] Fan Yang、Ryota Hinami、Yusuke Matsui、Steven Ly 和 Shin’ichi Satoh，《通过将扩散解耦为在线和离线处理实现高效图像检索》，载于*AAAI*人工智能会议论文集，2019年，第33卷，第9087–9094页。

[18] Tobias Weyand、Andre Araujo、Bingyi Cao 和 Jack Sim，《 Google 地标数据集 v2—— 用于实例级识别与检索的大规模基准》，载于*IEEE*计算机视觉与模式识别会议，2020年，第 2575–2584 页。

[19] James Philbin、Ondrej Chum、Michael Isard、Josef Sivic 和 Andrew Zisserman，《使用大型词汇表和快速空间匹配进行物体检索》，载于*2007*年 *IEEE* 计算机视觉与模式识别会议。IEEE，2007年，第 1–8 页。

[20] James Philbin、Ondrej Chum、MichaelIsard、Josef Sivic 和 Andrew Zisserman，《迷失于量化：改进大规模图像中的特定目标检索》

数据库》，载于*2008* 年 *IEEE* 计算机视觉与模式识别会议。IEEE，2008 年，第 1–8 页。

[21] Filip Radenovi´c、Ahmet Iscen、Giorgos Tolias、Yannis Avrithis 和 Ondˇrej Chum，“重新审视牛津和巴黎：大规模图像检索基准测试”，载于*IEEE*计算机视觉与模式识别会议论文集，2018年，第5706–5715页。

[22] Ali S Razavian、Josephine Sullivan、Stefan Carlsson 和 Atsuto Maki，“使用深度卷积网络进行视觉实例检索”，*ITE*媒体技术与应用学报，第4卷，第3期，第251–258页，2016年。

[23] Artem Babenko 和 Victor Lempitsky，“聚合局部深度特征以进行图像检索”，载于*IEEE*国际计算机视觉会议，2015年，第1269–1277页。

[24] Yannis Kalantidis、Clayton Mellina 和 Simon Osin- dero，“聚合深度卷积特征的跨维度加权”，载于欧洲计算机视觉会议，2016年，第685–701页。

[25]Maxim Berman、Herv´e J´egou、Andrea Vedaldi、Iasonas Kokkinos 和 Matthijs Douze，“Multigrain：面向类别和实例的统一图像嵌入”，*arXiv:1902.05509*，2019年。

[26] HeeJae Jun、Byungsoo Ko、Youngjoon Kim、Insik Kim 和 JongtackKim，“用于图像检索的多种全局描述符组合”，*arXiv:1903.10663*，2019年。

[27] Elad Hoffer 和 Nir Ailon，“使用三元组网络进行深度度量学习”，载于相似性模式识别国际研讨会。施普林格，2015年，第84–92页。

[28] Sivic 和 Zisserman，“Video Google：一种用于视频中对象匹配的文本检索方法”，载于*IEEE*计算机视觉国际会议，2003年，第2卷，第1470–1477页。

[29] Herv´e J´egou、Matthijs Douze、Cordelia Schmid 和 Patrick P´erez，“将局部描述符聚合为紧凑图像表示”，载于 *IEEE*计算机视觉与模式识别会议，2010年，第3304–3311页。

[30] Florent Perronnin、Yan Liu、Jorge S´anchez 和 Herv´e Poirier，“利用压缩费希尔向量进行大规模图像检索”，载于 *2010*年 *IEEE*计算机学会计算机视觉与模式识别会议，2010年，第3384–3391页。

[31] Relja Arandjelovic、Petr Gronat、Akihiko Torii、Tomas Pajdla 和 Josef Sivic，“Netvlad：用于弱监督地点识别的卷积神经网络架构”，载于 *IEEE*计算机视觉与模式识别会议，2016年，页码 5297–5307。[32] Maithra Raghu、Thomas Unterthiner、Simon Kornblith、Chiyuan Zhang 和 Alexey Dosovitskiy，“视觉 Transformer 看起来像卷积神经网络吗？”，神经信息处理系统进展，第 34 卷，2021年。[33] Andrew Brown、Weidi Xie、Vicky Kalogeiton 和 Andrew Zisserman，“Smooth-ap：为大规模图像检索铺平道路”，载于 欧洲计算机视觉会议。施普林格，2020年，页码 677–694。

[34] Relja Arandjelovi´c 和 Andrew Zisserman，“每个人都应了解的提升目标检索效果的三件事”，载于 *IEEE*计算机视觉与模式识别会议，2012年，第2911–2918页。

[35] Ondˇrej Chum、AndrejMikul´ık、Michal Perdoch 和 Jiˇr´ı Matas，“Total Recall II：重新审视查询扩展”，载于*IEEE*计算机视觉与模式识别会议，2011年，第889–896页。

[36] Ondrej Chum、James Philbin、Josef Sivic、Michael Isard 和 Andrew Zisserman，《Total Recall：用于目标检索的生成式特征模型与自动查询扩展》，载于*IEEE*计算机视觉会议，2007年11月，第 1–8 页。

[37] Xuanmeng Zhang、Minyue Jiang、Zhedong Zheng、Xiao Tan、Errui Ding 和 Yi Yang，《理解图像检索重新排序：图神经网络视角》，2020年。

[38] James M. Joyce，*Kullback–Leibler*散度，第 720–722 页，施普林格柏林海德堡，2011年。

[39] Jiankang Deng、Jia Guo、Niannan Xue 和 Stefanos Zafeiriou，《ArcFace：用于深度人脸识别的加性角度间隔损失》，载于*IEEE*计算机视觉与模式识别会议，2019年，第 4690–4699 页。

[40]Mathilde Caron、Ishan Misra、Julien Mairal、Priya Goyal、Piotr Bojanowski 和 Armand Joulin，“通过对比聚类分配进行视觉特征的无监督学习”，神经信息处理系统进展，第 33 卷，第 9912–9924 页，2020年。

[41] Stylianos Moschoglou、Athanasios Papaioannou、Chris- tos Sagonas、Jiankang Deng、Irene Kotsia 和 Stefanos Zafeiriou，“Agedb：首个手工收集的野外年龄数据库”，载于*IEEE*计算机视觉与模式识别会议研讨会，2017年，第51–59页。

[42]Tianyue Zheng、Weihong Deng 和 Jiani Hu，“跨年龄 LFW：用于研究非受限环境下跨年龄人脸识别的数据库”，*arXiv:1708.08197*，2017年。

[43] Cameron Whitelam、Emma Taborsky、Austin Blanton、Brianna Maze、Jocelyn Adams、Tim Miller、Nathan Kalka、Anil K Jain、James A Duncan、Kristen Allen 等人，“IARPA Janus Benchmark-B 人脸数据集”，载于*IEEE*计算机视觉与模式识别会议研讨会，2017年，第90–98页。

[44] James Philbin、Ondrej Chum、Michael Isard、Josef Sivic 和 Andrew Zisserman，“采用大型词汇表和快速空间匹配的物体检索”，载于*IEEE*计算机视觉与模式识别会议，2007年。

[45] James Philbin、Ondrej Chum、Michael Isard、Josef Sivic 和 Andrew Zisserman，“迷失于量化：改进大规模图像数据库中的特定目标检索”，载于*IEEE*计算机视觉与模式识别会议，2008年。[46]Zu Kim、Andr´e Araujo、Bingyi Cao、Cam Askew、Jack Sim、Mike Green、N’Mah Fodiatu Yilla 和 Tobias Weyand，“迈向更公平的地标识别数据集”，2021年。[47] Shuhei Yokoo、Kohei Ozaki、Edgar Simo-Serra 和 Satoshi Iizuka，“用于大规模地标检索的两阶段判别式重排序”，载于*IEEE*计算机视觉与模式识别会议研讨会，2020年，第1012–1013页。[48] Zhang Yuqi、Xu Xianzhe、Chen Weihua、Wang Yaohua、Zhang Fangyi、Wang Fan 和 Li Hao，“Google 地标检索 2021 第二名解决方案”，2021年。