

# 跨视角地理定位的硬负样本采样

## 摘要

跨视图地理定位是一项难题，需要额外的模块、特定的预处理或缩放策略来确定图像的准确位置。由于不同的视图具有不同的几何形状，因此像极坐标变换这样的预处理有助于合并它们。但是，这会导致图像失真，然后必须进行校正。在训练批次中添加硬负样本可以提高整体性能，但是在地理定位中使用默认的损失函数很难包含它们。作者提出了一种基于对比学习的简化但有效的架构，使用对称的InfoNCE损失，超越了当前的最先进的结果。网络框架由一个窄的训练管道组成，消除了使用聚合模块的需要，避免了进一步的预处理步骤，甚至增加了模型对未知区域的泛化能力。作者介绍了两种硬负采样策略。第一种显式地利用地理邻近位置来提供一个好的起点。第二种利用图像嵌入之间的视觉相似性来挖掘硬负样本。作者的工作在常见的跨视图数据集上表现出优异的性能，如CVUSA、CVACT。实验结果证明了他们的模型具有良好的泛化能力。

**关键词：**跨视角地理定位；硬负采样策略

## 1 引言

从图像中确定地理位置而不使用元数据是计算机视觉尚未解决的难题之一。解决这个问题可以帮助农业和汽车等领域。例如，农业中用于喷洒肥料的机器人需要高精度的位置。这可以通过实时GPS来实现，但这些传感器很昂贵，而且短时间的信号中断会阻碍工作流程。因此，在这些高度重复的环境中，基于航拍图像的定位可以进一步提高定位精度[4]。城市中的另一个挑战是所谓的城市峡谷效应，它阻挡了GPS等信号或降低了它们的准确性。特别是在大城市，GPS信号由于高楼大厦而受到干扰。Brosh等人对纽约市交通中的25万小时的驾驶评估显示，40%的GPS信号有10米的误差。因此，他们提出了一种基于图像检索的计算机视觉解决方案[2]来纠正这些信号。

传统的方法试图利用视觉线索来实现这一目标，例如太阳位置和产生的阴影[7, 15, 17]或天气[13, 14]，而当前的方法越来越侧重于基于深度学习的图像检索[1, 18, 19]。在跨视角图像检索中[33, 34, 39, 41]，图像的不同视角，例如地面图像和卫星图像，必须匹配以确定所搜索的位置。地面图像查询与具有已知地理位置的卫星图像数据库进行比较。

以前的方法主要使用基于CNN的方法[3, 11, 24–26, 31, 32, 34]，而目前的研究主要集中在使用Transformer或MLP Mixer架构作为地理定位的主干[36, 38, 40, 42]。在后一种情况下，只有当两个视角特定编码器之间的权重不共享时，才能实现合理的性能，从而导致模型变大。此外，极坐标变换经常被用来弥合视角之间的几何差距[24]。三元损失作为跨视角地理定位中的标准损失，每个批次只使用一个负例，当使用难负样本时，容易导致模型崩溃[35]。

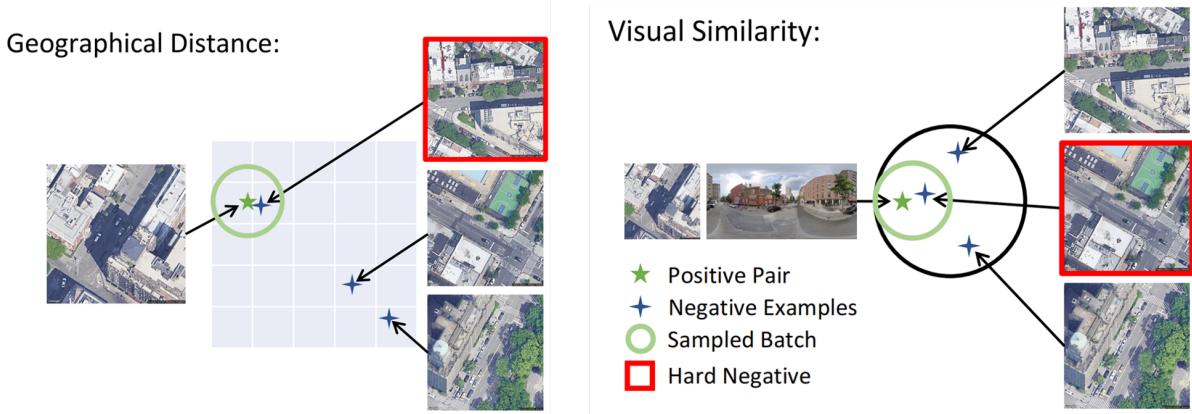


图 1. 提出了两种采样策略。第一种是基于卫星图像之间的地理距离。第二种是利用街景和卫星图像嵌入之间的余弦相似度，在一定范围内找出硬负样本。

作者提出了一个权重共享的孪生CNN网络，它基于InfoNCE损失学习与类无关的嵌入，并展示了CNN架构的优势 [5, 9, 21]。使用多模态预训练的技术对称地计算损失 [22]，以进一步帮助理解视点中的不同领域。在深度度量学习中，硬负样本，即模型难以区分的样本和正样本是实现卓越性能的关键因素，因此作者提出了两种采样方法。在前期，利用GPS坐标来对比邻近的地理邻居，作为第二次采样的良好初始化。在后来的批次中，收集视觉上相似的样本（例如余弦相似度）进行批量构建，以专注于硬负样本采样。总结一下在这项工作中的贡献：

- 展示了使用对称 InfoNCE 损失进行对比训练的优越性能。
- 提出了全球定位系统采样技术作为训练开始时针对特定任务的硬负样本采样技术。
- 提出了动态相似性采样 (DSS)，以根据街道和卫星视图之间的余弦相似性在训练过程中选择硬负样本。
- 提出的框架由一个简单的训练管道组成，无需特殊的聚合模块或复杂的预处理步骤，同时在性能和泛化能力方面优于当前的方法。

## 2 相关工作

Workman等 [34]的首批研究成果之一表明，CNN提取的特征远远优于手工制作的特征。在他们的工作中，还介绍了CVUSA数据集，这是当今地理定位的主要基准之一。为了将双流CNN网络训练为暹罗网络 [6, 27]，使用了一个简单的L2目标来减少视图特征之间的距离。在接下来的工作中，Zhai等人 [37]为鸟瞰图使用了嘈杂的语义标签，并最小化了这些标签之间的距离，并对街景进行了学习转换。特别是，这应该更好地反映视图中的通用语义布局。Vo等 [30]将软边际三重态损失作为跨视角地理定位的标准。三元组损失会减小正样本到锚点的距离，并增加到负样本的距离。Hu等 [11]的后续工作将NetVLAD层 [1]纳入其架构中，以进一步增强全局描述。输出的特征图是基于差异聚类进行聚合的，而不是基于简单的平均池或最大池。由于软边缘三重损失的收敛速度缓慢，他们提出了加权软边际排名损失，其中正样本和负样本之间的距离被额外缩放。

最初也注意到图像内部的方向，因为在像CVUSA [34]这样的数据集中，对齐是预先知道的，可以作为一个附加特征，如Liu等人 [19]所示。为了帮助网络更好地利用方向信息，扩展了输入通道的数量，并从上到下、从左到右使用了彩色编码的方向，同时用于街景和卫星景观。此外，他们还引入了CVACT数据集。

在随后的几篇文章 [3, 12, 25, 30] 中，该方向被进一步用于创建一个辅助目标。街景图像的偏移作为一种增强，偏移的程度必须被预测。

由于卫星图像和街景之间的几何形状不同，Shi等人 [24] 对卫星图像应用了极坐标变换。所得到的拉伸图像类似于街景图像的领域，这仍然是一种常见的预处理技术。此外，他们还使用空间感知特征聚合模块扩展了网络，以可学习的方式从特征映射中汇集重要特征。

极性变换的缺点是失真会给图像注入干扰。Toker et al. [28] 开发了一种方法，在GANs [8] 的帮助下学习消除这些干扰。将街景图像作为GAN鉴别器的地面真相。在推理时，生成器和鉴别器不再被使用，只使用潜在的表示来寻找与街景的相似性。

Yang等人 [36]介绍了使用ResNet主干与Transformer结合来完成该任务。由CNN输出的特征图提供了一个位置编码，然后输入到Transformer中。

由于前面提到的基准测试正在慢慢饱和，Zhu等人 [41] 创建了一个更具挑战性的活力数据集。而CVUSA、CVACT和VIGOR只使用街景图像查询卫星图像，郑等人创建了University-1652数据集 [39] 与无人机视图。

以前的方法致力于一个无法纠正的单一预测步骤，因此TransGeo [40] 和SIRNet [20] 强调了额外的细化工作。在SIRNet中，额外的细化模块被添加到CNN主干中。基于softmax的决策过程使用可变数量的模块，最多为4个。在TransGeo中，在其Transformer架构中使用注意图执行一个额外的缩放步骤。这导致较小的物体被以更高的分辨率观察。为了增加他们的方法的推广，TransGeo还使用自适应锐度感知最小化(ASAM) [16] 来平滑损失。ASAM显著减慢了训练过程，因为所有训练数据都需要额外的向前传递。由于几何视角被不同的视角强烈地移动，GeoDTR试图用基于Transformer的提取器提取额外的几何特性。首先，使用CNN网络独立生成特征。然而，由于这些几何特征的地面前真实标签缺失，一个额外的反事实损失被用来对比CNN的输出特征与几何提取器的输出特征，以确保差异。然后，在与视图相关的几何提取器的特征之间计算出一个三重态损失。Zhu等人 [5] 的另一种架构SAIG-D使用了MLPMixer [29]。他们用卷积骨干替换了补丁骨干，以支持局部特征学习。此外，还提出了一个特征聚合模块。与TransGeo类似，利用锐度感知最小化 [16] 来进一步平滑损失。

## 3 方法

### 3.1 方法概述

跨视图地理定位的深度度量学习使用默认的三联体损失 [23]，或者是使用各种扩展，如软边缘三联体损失 [30] 或加权软边缘三联体损失 [11]。三重态损失的目的是减少正例与锚之间的距离，同时增加负例与锚之间的距离。要使用三联体损失，必须事先取样合适的三联体。考虑到这一点，设计了框架来利用多重硬负样本。

### 3.2 对称信息NCE损失

对比学习的另一种方法，利用批次中所有可用的负值，即InfoNCE损失 [21,22]或NTXent损失 [5,9]，见公式1。

$$\mathcal{L}(q, R)_{\text{InfoNCE}} = - \log \frac{\exp(q \cdot r_+ / \tau)}{\sum_{i=0}^R \exp(q \cdot r_i / \tau)} \quad (1)$$

$q$ 表示一个编码的街景，即所谓的查询图像， $R$ 是一组被称为参考的编码卫星图像。只有一个正的 $r_i$ ，即 $r_+$ 匹配到 $q$ 。InfoNCE损失使用点积来计算查询图像和参考图像之间的相似性，当查询和正匹配相似时结果较低，当负 $r_i$ 与 $q$ 不相似时结果较高。计算交叉熵，作为视图之间相似性的损失函数。温度参数 $\tau$ 是一个超参数，可以学习或设置为一个静态值。到目前为止，InfoNCE损失主要以非对称的方式用于图像 [5,9]的无监督表示学习。对称公式在多模态预训练 [22]中显示是有用的，以弥补模态之间的差距。因此，以同样的对称方式利用这个损失函数来利用两个方向上的信息流：从卫星视图到街景，反之亦然。在InfoNCE损失中，一个正例总是与N-1个负例进行对比，其中N表示批次的大小，因此同时划分了许多例子。

### 3.3 模型体系结构

使用如图 2 所示的网络结构。利用平均值池化特征向量，无需任何注意力池化或细化模块。ConvNeXt 是ResNet [10]的变体，并使用了许多改进，例如修补主干、更高的内核大小、GELU 激活函数、LayerNorm 和深度卷积。虽然当前的工作由于不同的视图域而没有使用权重共享，但强调使用权重共享的 ConvNeXt 作为两个视图的单个编码器。

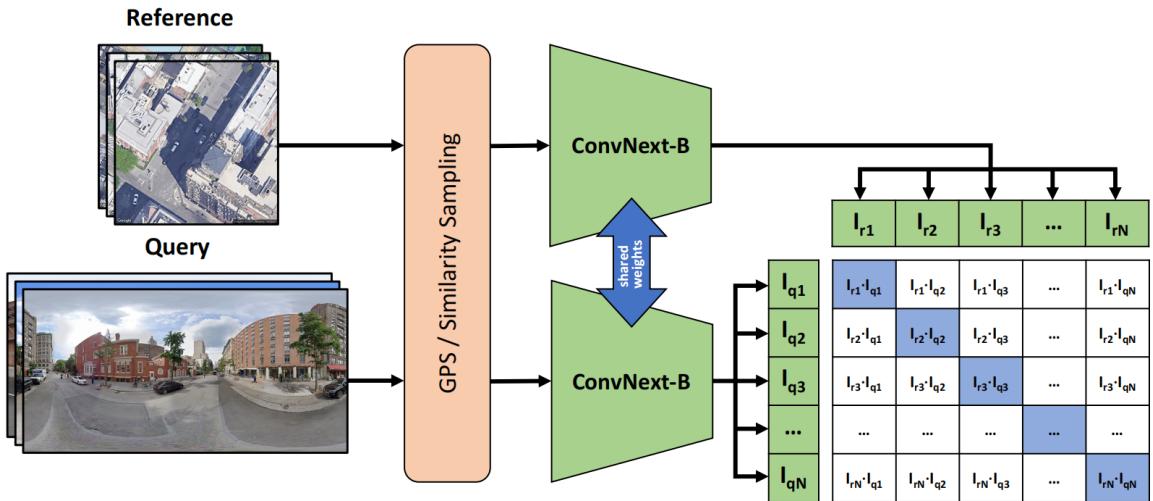


图 2. 体系结构概述。使用了一个现成的ConvNeXt-B网络，并基于GPS和视觉相似性采样挖掘硬负样本。InfoNCE损失以对称的方式使用来学习两个视图方向上的鉴别特征。

### 3.4 基于GPS的近邻采样方法

在大多数情况下，在训练之前无法选择困难示例，因为模型不适合训练数据的域。为了绕过这个缺点，利用跨视图地理定位的地理性质。来自同一区域甚至 VIGOR 数据集中同一条街道的图像具有共同的属性，例如植被、街道标志或住房类型。由于这些是对比乡村与城市景观的简单特征，因此它们不会对目标函数有太大贡献。因此，提出了一种简单的基于

GPS 的采样策略，用于在训练开始之前进行负样本挖掘初始化。对于 CVUSA 和 VIGOR，数据集中存在 GPS 位置，并且根据半正矢距离选择近邻。在 CVACT 数据集中，位置位于 UTM 坐标系中，这就是使用欧氏距离来确定邻居的原因。GPS坐标仅在前期时期的采样策略训练期间使用。

### 3.5 动态相似度采样

经过一定数量的轮次后，基于 GPS 的采样被动态相似性采样所取代。在训练数据的一个完整推理周期中，所有样本之间的视觉距离是使用余弦相似度计算的。为了对未来批次进行采样，选择每个查询图像的前 K 个最近邻。将 K 设置为小于或等于批量大小，在一批中包含多个地区或城市。这些 K 个邻居根据其相似性进行排序，然后为批次选择  $k/2$  个最近的样本，剩余的  $k/2$  个样本是从 K 中的剩余样本中随机选择的。随机选择过程确保了硬负例有足够的多样性，因为只计算每轮次的新距离，以缩短训练过程。超参数设置如下： $k = 64$ 、 $K = 128$  和  $e = 4$ 。在将 k 个样本添加到批次中之前，会进行查找以避免一个时期内出现重复条目。对于 k 的不同设置，即使  $k = K$ ，也没有观察到 [35] 中描述的模型崩溃问题。

## 4 代码介绍

这段代码定义了一个名为 ‘calculate\_nearest’ 的函数，用于在查询特征和参考特征之间执行最近邻检索，如图3所示。该函数接受多个输入参数并执行以下步骤：(1) 数据准备：‘query\_features’ 包含查询样本特征的张量。‘reference\_features’：包含参考样本特征的张量。‘query\_labels’：包含查询样本标签的张量。‘reference\_labels’：包含参考样本标签的张量。‘neighbour\_range’：要检索的最近邻的数量。‘step\_size’：处理查询特征时的每个步骤的大小。(2) 相似度计算：函数根据指定的 ‘step\_size’ 将查询特征分成块，并计算每个查询特征块与所有参考特征之间的余弦相似度。(3) Top-k 相似度分数：对于每个查询特征块，它识别前 k 个相似度分数及其在参考集中的相应索引。(4) 过滤邻居：对于每个查询样本，它提取了来自参考集的前 k 个邻居的标签。它创建了一个掩码，以过滤掉与查询样本具有相同标签的地面真相 (GT) 命中。(5) 构建最近邻字典：它创建了一个字典 (nearest\_dict)，其中键是查询标签，值是最近邻居标签的列表。根据相似度分数选择邻居，排除地面真相命中。(6) 输出：该函数返回包含每个查询样本的最近邻的 ‘nearest\_dict’。

```

def calculate_nearest(query_features, reference_features, query_labels, reference_labels, neighbour_range=64, step_size=1000):
    Q = len(query_features)
    steps = Q // step_size + 1
    similarity = []
    for i in range(steps):
        start = step_size * i
        end = start + step_size
        sim_tmp = query_features[start:end] @ reference_features.T
        similarity.append(sim_tmp.cpu())
    # matrix Q x R
    similarity = torch.cat(similarity, dim=0)
    topk_scores, topk_ids = torch.topk(similarity, k=neighbour_range+1, dim=1)
    topk_references = []
    for i in range(len(topk_ids)):
        topk_references.append(reference_labels[topk_ids[i], :])
    topk_references = torch.stack(topk_references, dim=0)
    # mask for ids without gt hits
    mask = topk_references != query_labels.unsqueeze(1)
    topk_references = topk_references.cpu().numpy()
    mask = mask.cpu().numpy()
    # dict that only stores ids where similarity higher than the lowest gt hit score
    nearest_dict = {}
    for i in range(len(topk_references)):
        nearest = topk_references[i][mask[i]][:neighbour_range]
        nearest_dict[query_labels[i].item()] = list(nearest)
    return nearest_dict

```

图 3. 查询特征和参考特征之间执行最近邻搜索

这段代码的主要功能是处理地理空间数据，计算基于欧氏距离的最近邻关系，并将结果保存到一个pickle文件中，如图4所示。下面是代码的详细介绍：(1) 导入模块和类：CVACTDatasetTrain一个自定义的类，来自sample4geo.dataset.cvact模块。这个类用于加载CVACT数据集的训练集。其他导入包括numpy、DistanceMetric（从sklearn.metrics导入）、scipy.io、pickle和torch。(2) 数据加载和准备：使用CVACTDatasetTrain类加载CVACT数据集的训练集，指定数据文件夹为”/hong614/dataset/CVACT”。从MAT文件（ACT\_data.mat）中加载UTM坐标和全景图像的ID。(3) 筛选训练集样本：通过遍历全景图像的ID，将属于训练集的样本的UTM坐标提取出来，并保存到utm\_coords字典中。同时，将样本的索引映射到数字索引并存储在train\_idsnum\_list中。(4) 计算欧氏距离矩阵：使用DistanceMetric计算UTM坐标之间的欧氏距离矩阵dm。(5) 转换为PyTorch Tensor：将NumPy数组dm转换为PyTorch Tensor，然后将对角线元素填充为矩阵中的最大值。(6) 获取最近邻关系：使用PyTorch的topk函数，获取每个样本的前TOP\_K个最近邻样本的索引和对应的欧氏距离值。(7) 保存最近邻关系：将最近邻关系保存到near\_neighbors字典中，其中键是样本的数字索引，值是最近邻样本的数字索引列表。将字典保存为Pickle文件（gps\_dict.pkl）。(8) 输出信息：打印输出一些有关训练集长度和计算结果的信息。

```

TOP_K = 128
dataset = CVACTDatasetTrain(data_folder = "/hong614/dataset/CVACT")
anuData = sio.loadmat('/hong614/dataset/CVACT/ACT_data.mat')
utm = anuData["utm"]
ids = anuData['panoIds']
idx2numidx = dataset.idx2numidx
train_ids_set = set(dataset.train_ids)
train_idsnum_list = []
utm_coords = dict()
utm_coords_list = []
for i, idx in enumerate(ids):
    idx = str(idx)
    if idx in train_ids_set:
        coordinates = (float(utm[i][0]), float(utm[i][1]))
        utm_coords[idx] = coordinates
        utm_coords_list.append(coordinates)
        train_idsnum_list.append(idx2numidx[idx])
print("Length Train Ids:", len(utm_coords_list))
train_idsnum_lookup = np.array(train_idsnum_list)
print("Length of gps coords : " + str(len(utm_coords_list)))
print("Calculation...")
dist = DistanceMetric.get_metric("euclidean")
dm = dist.pairwise(utm_coords_list, utm_coords_list)
print("Distance Matrix:", dm.shape)
dm_torch = torch.from_numpy(dm)
dm_torch = dm_torch.fill_diagonal_(dm.max())
values, ids = torch.topk(dm_torch, k=TOP_K, dim=1, largest=False)
values_near_numpy = values.numpy()
ids_near_numpy = ids.numpy()
near_neighbors = dict()
for i, idnum in enumerate(train_idsnum_list):
    near_neighbors[idnum] = train_idsnum_lookup[ids_near_numpy[i]].tolist()
print("Saving...")
with open("/hong614/dataset/CVACT/gps_dict.pkl", "wb") as f:
    pickle.dump(near_neighbors, f)

```

图 4. 计算图像之间的最近邻关系。

这段代码执行了极坐标变换（polar transformation）操作，将原始航拍图像转换为极坐标形式，并保存转换后的图像，如图5所示。以下是代码的详细介绍：(1) 参数设置：‘S’：原始航拍图像的大小。‘height’：极坐标变换后图像的高度。‘width’：极坐标变换后图像的宽度。(2) 坐标计算：使用NumPy的arange函数生成一维数组i和j，表示高度和宽度上的索引。使用meshgrid函数创建二维坐标网格jj和ii，表示图像中每个像素的坐标。(3) 极坐标变换计算：根据极坐标变换的公式，计算新的坐标x和y。这里使用了正弦和余弦函数，以及图像的

高度和宽度信息。(4) 输入输出目录设置：指定原始图像的输入目录和极坐标变换后图像的输出目录。如果输出目录不存在，则创建输出目录。(5) 遍历图像并进行极坐标变换：获取输入目录中的所有图像文件。对于每个图像文件，读取原始图像，然后使用sample\_bilinear函数对图像进行双线性采样，根据计算得到的极坐标坐标x和y进行变换。变换后的图像保存到输出目录中，文件格式由'jpg'替换为'png'。

```
#####
# Apply Polar Transform to Aerial Images in CVUSA Dataset #####
S = 750 # Original size of the aerial image
height = 112 # Height of polar transformed aerial image
width = 616 # Width of polar transformed aerial image

i = np.arange(0, height)
j = np.arange(0, width)
jj, ii = np.meshgrid(j, i)

y = S/2. - S/2./height*(height-1-ii)*np.sin(2*np.pi*jj/width)
x = S/2. + S/2./height*(height-1-ii)*np.cos(2*np.pi*jj/width)

input_dir = '../Data/CVUSA/bingmap/19/'
output_dir = '../Data/CVUSA/polarmap/19/'

if not os.path.exists(output_dir):
    os.makedirs(output_dir)

images = os.listdir(input_dir)

for img in images:
    signal = imread(input_dir + img)
    image = sample_bilinear(signal, x, y)
    imsave(output_dir + img.replace('jpg', 'png'), image)
```

图 5. 极坐标变换操作。

这段代码实现了双线性插值的操作，如图6所示。双线性插值是一种在离散的二维网格上估算非整数坐标位置的像素值的方法。以下是代码的详细介绍：(1)输入参数：‘signal’：输入信号，通常是一个图像或图像的一部分。‘rx’：x方向上的插值坐标。‘ry’：y方向上的插值坐标。(2) 获取输入信号的维度：通过signal.shape获取输入信号的维度，分别存储在signal\_dim\_x和signal\_dim\_y中。(3) 计算四个样本坐标：将浮点型插值坐标‘rx’和‘ry’转换为整数型坐标‘ix0’、‘iy0’、‘ix1’、‘iy1’。这四个坐标分别代表原图中的四个采样点。(4) 处理越界情况：定义图像的边界范围为bounds，防止采样坐标超出图像范围。使用sample\_within\_bounds函数在图像边界内对四个采样点进行采样。(5) 在四个位置进行双线性插值：分别在四个位置（左上、右上、左下、右下）对信号进行采样，得到‘signal\_00’、‘signal\_10’、‘signal\_01’、‘signal\_11’。在x方向进行线性插值：使用线性插值计算在x方向上的插值结果‘fx1’和‘fx2’，其中‘fx1’表示在‘ix1’位置处的插值结果，‘fx2’表示在‘ix0’位置处的插值结果。(7) 在y方向进行线性插值：最终的插值结果是在y方向上对‘fx1’和‘fx2’进行线性插值，得到最终的双线性插值结果。(8) 返回结果：返回经过双线性插值后的像素值。

```

def sample_bilinear(signal, rx, ry):

    signal_dim_x = signal.shape[0]
    signal_dim_y = signal.shape[1]

    # obtain four sample coordinates
    ix0 = rx.astype(int)
    iy0 = ry.astype(int)
    ix1 = ix0 + 1
    iy1 = iy0 + 1

    bounds = (0, signal_dim_x, 0, signal_dim_y)

    # sample signal at each four positions
    signal_00 = sample_within_bounds(signal, ix0, iy0, bounds)
    signal_10 = sample_within_bounds(signal, ix1, iy0, bounds)
    signal_01 = sample_within_bounds(signal, ix0, iy1, bounds)
    signal_11 = sample_within_bounds(signal, ix1, iy1, bounds)

    na = np.newaxis
    # linear interpolation in x-direction
    fx1 = (ix1-rx)[...,na] * signal_00 + (rx-ix0)[...,na] * signal_10
    fx2 = (ix1-rx)[...,na] * signal_01 + (rx-ix0)[...,na] * signal_11

    # linear interpolation in y-direction
    return (iy1 - ry)[...,na] * fx1 + (ry - iy0)[...,na] * fx2

```

图 6. 双线性插值操作。

这段代码实现了在给定边界范围内采样输入信号的功能，如图7所示。以下是代码的详细介绍：(1) 输入参数：‘signal’：输入信号，通常是一个图像或图像的一部分。x: x方向上的采样坐标。y: y方向上的采样坐标。‘bounds’：边界范围，包括‘xmin’、‘xmax’、‘ymin’和‘ymax’。(2) 解析边界范围：从bounds中解析出xmin、xmax、ymin和ymax。(3) 检查采样点是否在边界内：使用布尔索引检查每个采样点是否在给定的边界范围内，生成一个布尔数组idxs。(4) 创建采样结果数组：创建一个与输入信号具有相同通道数的零数组，表示采样结果。数组的形状是(x.shape[0], x.shape[1], signal.shape[-1])，其中x.shape[0]和x.shape[1]分别是x和y的形状。(5) 将边界内的采样点赋值到采样结果数组：使用布尔索引‘idxs’，将输入信号中在边界内的采样点的值赋值给采样结果数组。(6) 返回采样结果：返回采样结果数组，其中包含了在给定边界范围内的输入信号的采样值。

```

def sample_within_bounds(signal, x, y, bounds):

    xmin, xmax, ymin, ymax = bounds

    idxs = (xmin <= x) & (x < xmax) & (ymin <= y) & (y < ymax)

    sample = np.zeros((x.shape[0], x.shape[1], signal.shape[-1]))
    sample[idxs, :] = signal[x[idxs], y[idxs], :]

return sample

```

图 7. 在给定边界范围内采样输入信号。

## 5 复现细节

### 5.1 与已有开源代码对比

复现的论文已经开源。我们在该代码的基础上对数据进行了极坐标变换处理，极坐标变换处理代码参考文献 [24]。

### 5.2 实验环境搭建

该实验旨在使用 Python 进行跨视角地理定位，并使用 pytorch 框架来建立和训练模型。在系统要求方面，我们选择Linux操作系统，使用Python 3.7版本。我们使用的服务器型号为NVIDIA A100-SXM4-40GB。

### 5.3 创新点

正如我们所观察到的，航拍图像中位于相同方位角方向上的像素大约对应于地面视图图像中的垂直图像列。我们没有强制神经网络隐式地学习这种映射，而是显式地变换航空图像，然后粗略地消除这两个域之间的几何对应间隙。这样做，简化了学习多个对应关系（即几何和特征表示）的任务，并且只需要学习一个简单的特征映射任务，从而显着促进网络收敛。

将极坐标变换应用于航空图像，以便在航空图像和地面图像之间建立更明显的空间对应关系。具体来说，将每张航拍图像的中心作为极坐标原点，将北向（卫星图像通常可用）作为极坐标变换中的  $0^\circ$  角。航拍图像没有临时的预居中过程，并且我们在测试期间不假设查询地面图像的位置对应于航拍图像的中心。事实上，极坐标原点上的小偏移不会严重影响极坐标变换航空图像的外观，并且减少小的外观变化。相反，当发生较大偏移时，航拍图像应被视为负样本，并且极坐标变换后的航拍图像将与地面实况图像显着不同。通过这种方式，极坐标变换有效地增加了我们模型的辨别力。

为了便于训练两分支网络，将变换后的航空图像的大小限制为与地面图像相同  $W_g * H_g$ 。需要注意的是，原始航拍图像的尺寸为  $A_a * A_a$ 。因此，原始航拍图像点  $(x_i^s, y_i^s)$  和目标变换航

拍图像点 $(x_i^t, y_i^t)$ 之间的极坐标变换定义为:

$$x_i^s = \frac{A_a}{2} + \frac{A_a}{2} \frac{y_i^t}{H_g} \sin\left(\frac{2\pi}{W_g} x_i^t\right) \quad (2)$$

$$y_i^s = \frac{A_a}{2} - \frac{A_a}{2} \frac{y_i^t}{H_g} \cos\left(\frac{2\pi}{W_g} x_i^t\right) \quad (3)$$

极坐标变换后, 变换后的航拍图像中的物体与地面图像中的物体处于相似的位置, 如图8所示。然而, 由于极坐标变换没有考虑场景内容的深度, 因此变换后的图像中的外观扭曲仍然很明显。减少图像描述符提取的这些失真伪影也是可取的。

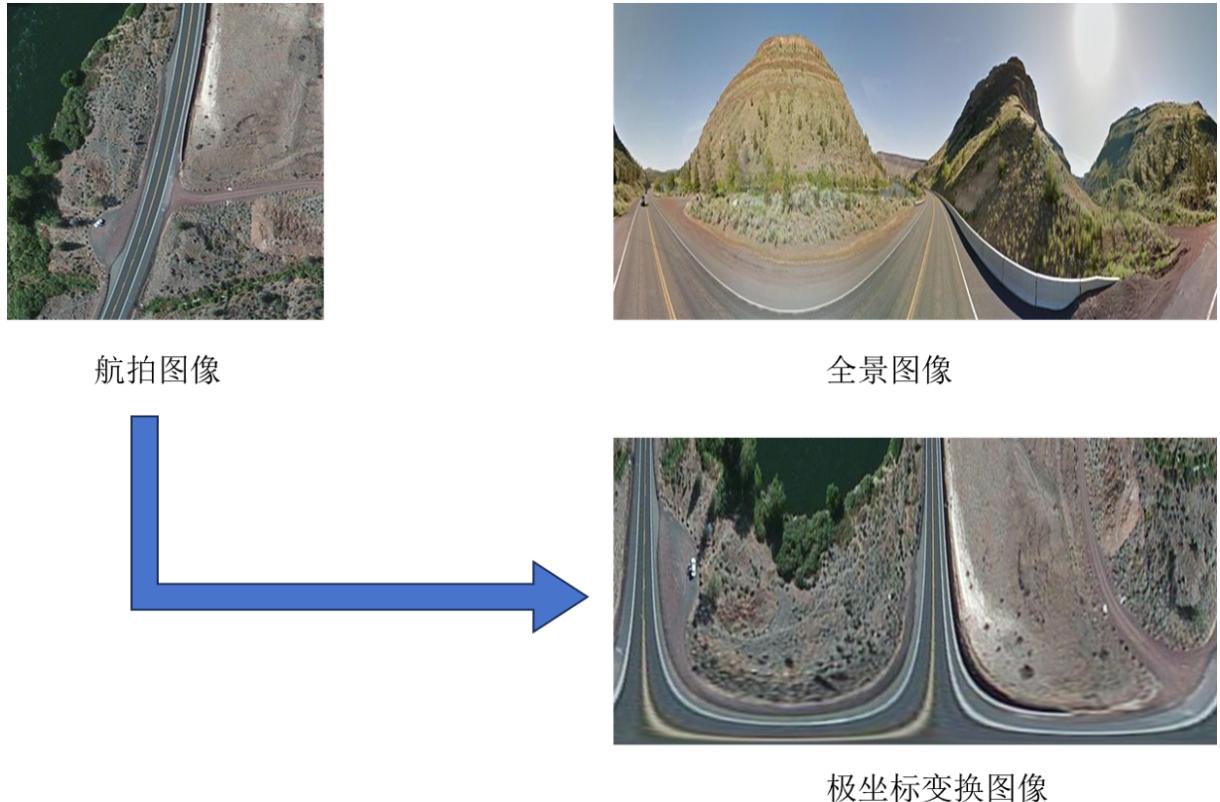


图 8. 实验结果示意

## 6 实验

### 6.1 数据集

我们的实验在两个标准基准数据集上进行: CVUSA 和 CVACT, 其中地面图像为全景图像。CVUSA 和 CVACT 都是跨视角数据集, 每个数据集包含 35 532 对地面和空中图像用于训练。CVUSA 提供 8 884 幅图像对用于测试, CVACT 提供相同数量的图像对用于验证 (记为 CVACT\_val)。此外, CVACT 还提供了 92 802 对带有准确地理标记的交叉视图图像来评估地理定位性能 (记为 CVACT\_test)。CVACT\_test 是一个真实的地理定位检索测试集, 所有与查询的地面图像相距 5 米以内的航空图像都被视为该查询图像的地面真实对应图像, 换句话说, 对于一个查询的地面图像, 数据库中可能存在多个对应的航空图像。请注意, 在这两个数据集中, 地面图像和航空图像是在不同时间拍摄的。

## 6.2 评价指标

评估指标：我们使用前 K 位的召回准确率作为评估指标来检验我们模型的性能。具体来说具体来说，给定一张地面查询图像，如果它的地面实况航拍图像在最近的K个范围内，则视为”成功定位”。航空图像在最近的前 K 幅检索图像之内，则该图像被视为”定位成功”。被正确定位的查询图像的百分比报告为  $r@K$ 。

## 6.3 实施细节

在我们的实验中，架构没有改变，使用的是具有 88M 参数的 ConvNeXt-B。我们在 InfoNCE 损失中使用了 0.1 的标签平滑，温度参数  $\tau$  是一个可学习的参数。正如 Zhang 等人所指出的，为了不干扰图像中编码的位置和方向，同步增强非常重要。因此，我们使用同步水平翻转和旋转卫星图像并相应地移动街景图像，以保持卫星图像的方向。每个实验都使用 AdamW 优化器进行了 40 轮的训练，批量大小为 128，初始学习率为 0.001，。

## 7 实验结果分析

我们将改进的方法和论文提供的结果和我们复现的结果进行了比较。在我们改进的方法中，我们对航拍图像进行极坐标变换。我们汇报了  $R@1$ ,  $R@5$ ,  $R@10$  和  $R1\%$  的召回率。结果列于表1。如表1所示，我们复现的原论文的结果和作者提供的相差不大，在CVUSA数据集上， $R@1$ 比原论文提供的减少0.31， $R@5$ 比原论文提供的减少0.14， $R@10$ 比原论文提供的减少0.06， $R1\%$ 比原论文提供的减少0.01。在CVACT数据集上， $R@1$ 比原论文提供的减少0.77， $R@5$ 比原论文提供的减少0.65， $R@10$ 比原论文提供的减少0.43， $R1\%$ 比原论文提供的提高了0.14。在我们改进的方法中，我们对航拍图像进行了极坐标变换，从而减小了航拍视图和全景视图之间的域差距。在CVUSA数据集上， $R@1$ 比原论文提供的减少0.13， $R@5$ 达到了和原论文一样的结果， $R@10$ 比原论文提供的减少0.02， $R1\%$ 比原论文提供的提高了0.02。在CVACT数据集上， $R@1$ 比原论文提供的减少0.66， $R@5$ 比原论文提供的提高了0.13， $R@10$ 比原论文提供的减少0.07， $R1\%$ 比原论文提供的提高了0.05。虽然说极坐标变换减小了两个视角之间的域差距，但是还是存在一些缺点：(1) 相对于直角坐标系，极坐标系的运算和表达相对较为复杂，特别是在涉及到距离、方向和旋转等计算时，可能需要更多的数学操作。(2) 从其他坐标系转换到极坐标，都涉及到复杂的数学运算，可能会引入一定的计算开销，特别是在实时定位和导航等应用中可能会受到性能的制约。

表 1. Results on Flickr30K test set in terms of  $R@K(\%)$

Methods	CVUSA				CVACT			
	R1	R5	R10	R1%	R1	R5	R10	R1%
Sample4Geo	98.68	99.68	99.78	99.87	90.81	96.74	97.48	98.77
复现 Sample4Geo	98.37	99.54	99.72	99.86	90.04	96.09	97.05	98.63
ours	98.55	99.68	99.76	99.89	90.15	96.87	97.41	98.72

此外，我们将改进的模型在CVUSA数据集和CVACT数据集上的图像可视化，如图9所示。可以看出，这些极其困难的范例在外观上非常相似。由于我们将航拍图像进行极坐标

变换从而减小了航拍视图和全景视图之间的域差距，从而获得更好的性能。

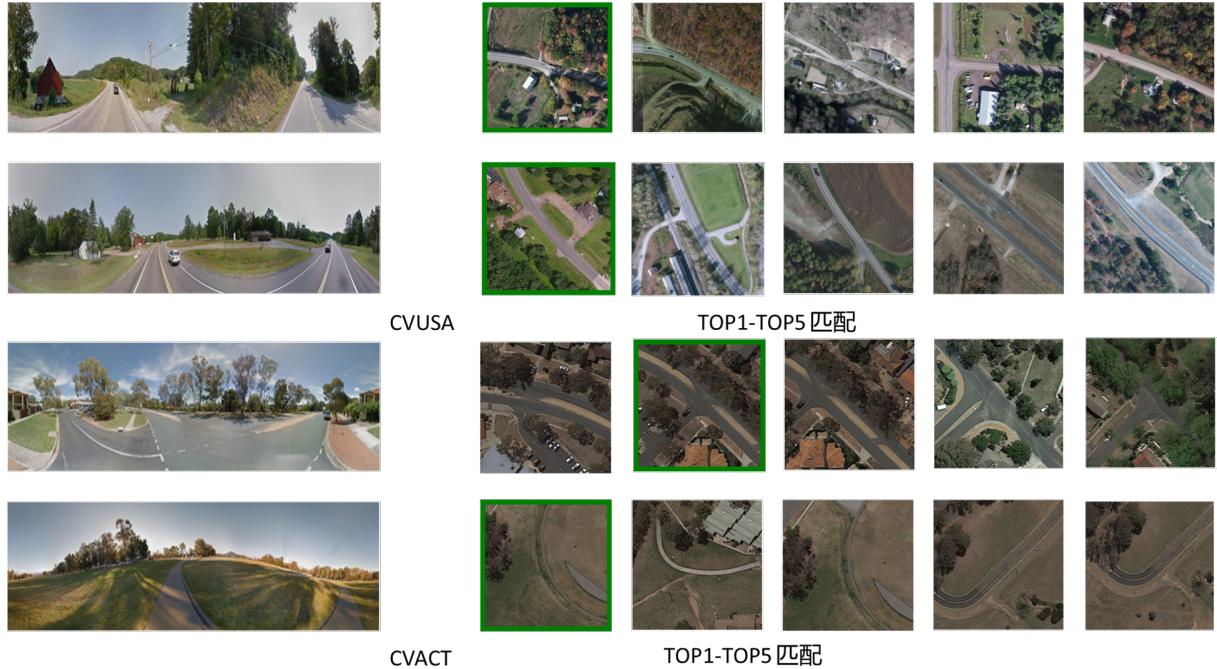


图 9. 在CVUSA和CVACT测试数据集上的检索结果的可视化。前5个结果从左到右排序，地面真实结果显示在绿色的方框中。

我们给出了改进的模型在CVUSA数据集上的热力图。我们观察到改进的方法更关注道路、路基、树木等上下文信息，这些信息更能合理地交叉查看地理定位。

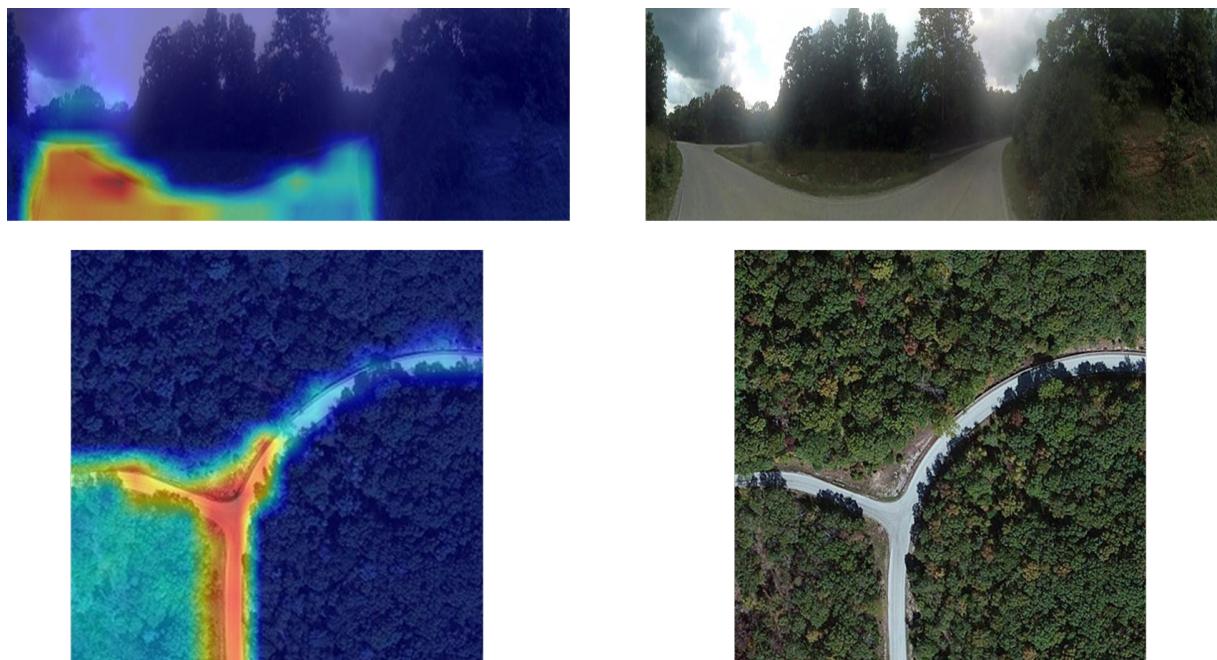


图 10. 可视化CVUSA数据集的热力图

## 8 总结与展望

在我们的工作中，我们提出了一个简单但有效的方法来解决地理定位任务。我们提出的模型由一个基于CNN的卫星和地面视图的单一图像编码器组成。这种轻量级的方法通过使用信息损失作为训练目标来利用对比学习。我们进一步证明，一个有效的采样策略可以在CVUSA和CVACT数据集上获得更好的结果。最后，由于我们将航拍图像进行极坐标变换从而减小了航拍视图和全景视图之间的域差距，从而获得更好的性能。

以前的数据集的另一个共同特征是重点关注城市环境。CVUSA在这里提供了更广的范围，但地面视图图像显然都是在街道上拍摄的。这并没有反映出野外场景的多样性。未来的数据集还应包括不完全从道路上获取的地面视图，以增加多样性和实用性。

## 参考文献

- [1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.
- [2] Eli Brosh, Matan Friedmann, Ilan Kadar, Lev Yitzhak Lavy, Elad Levi, Shmuel Rippa, Yair Lempert, Bruno Fernandez-Ruiz, Roei Herzig, and Trevor Darrell. Accurate visual localization for automotive applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
- [3] Sudong Cai, Yulan Guo, Salman Khan, Jiwei Hu, and Gongjian Wen. Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8391–8400, 2019.
- [4] Nived Chebrolu, Philipp Lottes, Thomas Läbe, and Cyrill Stachniss. Robot localization based on aerial images for precision agriculture tasks in crop fields. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1787–1793. IEEE, 2019.
- [5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [6] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, volume 1, pages 539–546. IEEE, 2005.
- [7] Fabio Cozman and Eric Krotkov. Robot localization using a computer vision sextant. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 1, pages 106–111. IEEE, 1995.

- [8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- [9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [11] Sixing Hu, Mengdan Feng, Rang MH Nguyen, and Gim Hee Lee. Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7258–7267, 2018.
- [12] Wenmiao Hu, Yichen Zhang, Yuxuan Liang, Yifang Yin, Andrei Georgescu, An Tran, Hannes Kruppa, See-Kiong Ng, and Roger Zimmermann. Beyond geo-localization: fine-grained orientation of street-view images by cross-view matching with satellite imagery. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6155–6164, 2022.
- [13] Nathan Jacobs, Kylia Miskell, and Robert Pless. Webcam geo-localization using aggregate light levels. In *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 132–138. IEEE, 2011.
- [14] Nathan Jacobs, Nathaniel Roman, and Robert Pless. Toward fully automatic geo-location and geo-orientation of static outdoor cameras. In *2008 IEEE Workshop on Applications of Computer Vision*, pages 1–6. IEEE, 2008.
- [15] Imran N Junejo and Hassan Foroosh. Gps coordinates estimation and camera calibration from solar shadows. *Computer Vision and Image Understanding*, 114(9):991–1003, 2010.
- [16] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pages 5905–5914. PMLR, 2021.
- [17] Jean-François Lalonde, Srinivasa G Narasimhan, and Alexei A Efros. What do the sun and the sky tell us about the camera? *International Journal of Computer Vision*, 88:24–51, 2010.
- [18] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5007–5015, 2015.
- [19] Liu Liu and Hongdong Li. Lending orientation to neural networks for cross-view geo-localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5624–5633, 2019.

- [20] Xiufan Lu, Siqi Luo, and Yingying Zhu. It’s okay to be wrong: Cross-view geo-localization with step-adaptive iterative refinement. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022.
- [21] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [23] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.
- [24] Yujiao Shi, Liu Liu, Xin Yu, and Hongdong Li. Spatial-aware feature aggregation for image based cross-view geo-localization. *Advances in Neural Information Processing Systems*, 32, 2019.
- [25] Yujiao Shi, Xin Yu, Dylan Campbell, and Hongdong Li. Where am i looking at? joint location and orientation estimation by cross-view matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4064–4072, 2020.
- [26] Yujiao Shi, Xin Yu, Liu Liu, Tong Zhang, and Hongdong Li. Optimal feature transport for cross-view image geo-localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11990–11997, 2020.
- [27] Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.
- [28] Aysim Toker, Qunjie Zhou, Maxim Maximov, and Laura Leal-Taixé. Coming down to earth: Satellite-to-street view synthesis for geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6488–6497, 2021.
- [29] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.
- [30] Nam N Vo and James Hays. Localizing and orienting street views using overhead imagery. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 494–509. Springer, 2016.

- [31] Tingyu Wang, Zhedong Zheng, Chenggang Yan, Jiyong Zhang, Yaoqi Sun, Bolun Zheng, and Yi Yang. Each part matters: Local patterns facilitate cross-view geo-localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(2):867–879, 2021.
- [32] Tingyu Wang, Zhedong Zheng, Zunjie Zhu, Yuhan Gao, Yi Yang, and Chenggang Yan. Learning cross-view geo-localization embeddings via dynamic weighted decorrelation regularization. *arXiv preprint arXiv:2211.05296*, 2022.
- [33] Scott Workman and Nathan Jacobs. On the location dependence of convolutional neural network features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 70–78, 2015.
- [34] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocation with aerial reference imagery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3961–3969, 2015.
- [35] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2840–2848, 2017.
- [36] Hongji Yang, Xiufan Lu, and Yingying Zhu. Cross-view geo-localization with layer-to-layer transformer. *Advances in Neural Information Processing Systems*, 34:29009–29020, 2021.
- [37] Menghua Zhai, Zachary Bessinger, Scott Workman, and Nathan Jacobs. Predicting ground-level scene layout from aerial imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 867–875, 2017.
- [38] Xiaohan Zhang, Xingyu Li, Waqas Sultani, Chen Chen, and Safwan Wshah. Geodtr+: Toward generic cross-view geolocation via geometric disentanglement. *arXiv preprint arXiv:2308.09624*, 2023.
- [39] Zhedong Zheng, Yunchao Wei, and Yi Yang. University-1652: A multi-view multi-source benchmark for drone-based geo-localization. In *Proceedings of the 28th ACM international conference on Multimedia*, pages 1395–1403, 2020.
- [40] Sijie Zhu, Mubarak Shah, and Chen Chen. Transgeo: Transformer is all you need for cross-view image geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1162–1171, 2022.
- [41] Sijie Zhu, Taojiannan Yang, and Chen Chen. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3640–3649, 2021.
- [42] Yingying Zhu, Hongji Yang, Yuxin Lu, and Qiang Huang. Simple, effective and general: A new backbone for cross-view image geo-localization. *arXiv preprint arXiv:2302.01572*, 2023.