

遥感中土地利用土地覆盖分割方法的比较

Naman Srivastava, Joel D Joy, Yash Dixit, Swarup E, Rakshit Ramesh
Center of Data for Public Good
Indian Institute of Science, Bengaluru

Abstract

土地利用和土地覆盖 (LULC) 制图对城市和资源规划至关重要, 是打造智能和可持续城市的关键要素之一。本研究评估了先进的 LULC 制图技术, 重点关注应用于 Cartosat 多光谱 (MX) 传感器图像的基于查找表 (LUT) 的大气校正, 然后是用于 LULC 预测的监督和半监督学习模型。我们探讨了 DeeplabV3+ 和交叉伪监督 (CPS)。CPS 模型进一步通过动态加权进行优化, 增强训练期间伪标签的可靠性。这种综合方法分析了 LULC 制图技术在各种城市规划应用中的准确性和效用。以印度海德拉巴为案例研究, 展现了快速城市化引起的显著土地利用变迁。通过分析不同时间段 Cartosat MX 图像, 我们突出了城市扩张、绿色空间缩小和工业区扩大的变化。这表明这些技术对城市规划师和政策制定者具有实用价值。

1 介绍

高分辨率的卫星多光谱图像在过去的二十年里已在包括国防、资源管理、环境分析和城市规划等众多行业中出现。这些图像由于其精细的空间分辨率覆盖和大传感器视场覆盖广泛区域而被视为了解和管理地球表面而不是基于实地采样的一个很好的替代方案 ([?])。然而, 为特定任务手动标注此类图像是费时且耗力的。为了解决这一挑战, 深度学习技术如目标检测和图像分割被用来自动化和增强地理空间数据的分析。在这些任务中, 土地利用土地覆盖 (LULC) 制图已经获得了重要意义, 使政府和组织能够监控资源、评估环境变化并制定明智的政策。深度学习算法可以提高 LULC 制图的准确性和效率, 为可持续发展和资源管理做出贡献。然而, 在将多光谱图像应用于深度学习之前, 还需要解决某些问题。原始多光谱图像通常以大气顶层 (TOA) 数字编号 (DN) 值

捕获, 受到诸如气溶胶、水汽和臭氧引起的散射和吸收等大气效应的影响, 这些影响污染了来自大气底层 (BOA) 表面的信号 [?]。为了在深度学习应用中准确地表征表面, 纠正这些大气干扰是至关重要的。因此, 使用大气校正 (AC) 将 TOA DN 值转换为 BOA 反射率对于确保数据分析就绪 (ARD) 以及提高预测准确性至关重要。在土地利用和土地覆盖 (LULC) 预测的深度学习模型中, 语义分割发挥了重要作用, U-Net 和 DeepLab v3+ 等架构被广泛使用。U-Net 最初是为生物医学图像分割而设计的, 具有编码器-解码器结构和残差连接的 U 型架构。该设计对于需要精确特征定位的任务非常有效。相比之下, 由 Google DeepMind 开发的 DeepLab v3+ 利用扩张卷积来扩展视野, 捕捉多尺度的上下文信息。这使得它特别擅长识别各种地貌类型, 如城市区域、森林、水体和农田。半监督分割在遥感中已被广泛研究, 包括一致性正则化和迭代训练等技术。半监督学习的另一种流行方法是交叉伪监督 (CPS), 它通过交叉监督在两个网络之间强制一致性以提高性能。然而, CPS 面临过拟合和类别不平衡等挑战。虽然一些研究已经应用了半监督技术来进行 LULC 预测, 并解决类别不平衡和地理环境 (例如, 沿海地区、农田) 等问题, 强大的 LULC 映射模型必须考虑因一天中的时间、季节以及印度不同气候和环境条件下的建筑和植被外观差异造成的卫星图像变化。一个通用模型应能在各种地形上表现良好, 如海岸、平原、农田、河流盆地和大都市地区。双去偏异质协同训练 (DHC) 框架引入了自适应损失加权策略: 分布感知去偏加权 (DistDW) 和难度感知去偏加权 (DiffDW)。DistDW 根据批次分布调整类别权重, 而 DiffDW 则根据骰子系数的变化动态加权类别。这些策略有效地解决了偏差和类别不平衡问题。进一步的改进由王等人提出。[?] 将这些策略整合到一个通用的半监督框架中, 解决了类别不平衡、无监督域适应和域泛化的问题。在大多数半监督学习方法中, 使用的是标记和未标记

的图像集。尽管已经存在许多标记准确且密集的遥感数据集 [?, ?, ?]，但它们在分辨率上有所不同。用户通常需要在与基准数据集不同分辨率的图像上训练其土地利用和土地覆盖 (LULC) 模型，这可能导致性能不佳。为此类情况准备地面真实标签通常需要手动标注或利用像 OpenStreetMap 这样的开放数据来源 [?]。然而，由于持续的建设和拆建，这些来源可能会受到标签稀疏、错误分类和信息过时的影响。有研究应用了弱监督分割技术用于土地利用土地覆盖 [?, ?, ?, ?]。在本研究中，我们应用大气校正技术，随后是流行的监督和半监督学习技术，并在一个标签稀疏的数据集上评估其性能。

2 方法论

提出的模型架构涉及几个关键阶段：从矢量数据准备栅格掩膜，将大型卫星图像划分为较小的补丁以进行模型训练，并训练模型以预测分割输出。最后，评估模型的性能以评估其在预测土地利用和土地覆盖类别掩膜方面的效果。图 1 展示了机器学习操作流程中这些阶段的顺序。

2.1 数据来源

2.1.1 栅格数据

本研究使用了高分辨率的多光谱影像 (约为 $1.134 \text{ m}^2/\text{px}$)，覆盖蓝光 ($0.45\text{-}0.52 \mu\text{m}$)、绿光 ($0.52\text{-}0.59 \mu\text{m}$)、红光 ($0.62\text{-}0.68 \mu\text{m}$) 及近红外光 ($0.77\text{-}0.86 \mu\text{m}$) 光谱波段。这些场景由印度空间研究组织 (ISRO) 运行的 CartoSAT-3 卫星上的多光谱 (MX) 传感器获取，覆盖了六个印度城市。为了确保土地使用和地表覆盖 (LULC) 类别的充分异质性，每个图块在预处理工作流的一部分中被裁剪成更小的子集 (详见图 1)。每个场景覆盖特定区域，云覆盖率低于 20%，但受传感器噪声和大气相互作用的影响，主要是由于气体的瑞利散射和气溶胶的吸收 (详见第 2.2 节)。各个波段被整合为 Zarr 格式的栅格堆栈 [?]，而非 Xarray [?]，因为它能够分别保存和加载数据块，优化内存使用。

2.1.2 矢量数据

训练和评估的矢量数据来源如下：

训练：来自班加罗尔、孟买、浦那、瓦拉纳西和德里的矢量数据是使用 JavaOpenStreetMap (JOSM) 编辑

器从 OpenStreetMap (OSM) [?] 获得的。这些数据被保存为在相应栅格文件地理空间边界内的 GeoJSON 文件。

评估：评估数据包括来自微软、谷歌和 JOSM 的预测建筑矢量数据，这些数据经过筛选以匹配海得拉巴的兴趣区域 (AOI)。这些矢量与 ISRO 的栅格图像对齐，去除冗余数据后被保存为 GeoJSON 文件。

数据准备过程包括生成特定类别的二值掩码，然后将它们合并成一个综合的多类别掩码。最初，使用共享的坐标参考系统 (CRS) 对齐矢量数据和相应的栅格文件。采用了各种矢量几何：道路使用线状图形 (Linestrings)，建筑物和水体使用多边形/多重多边形 (Polygons/Multipolygons)，每种几何用于生成特定类别的真实掩码，处理方式各不相同。对于道路，使用 Shapely 库对线状图形应用了 3 像素的缓冲，而建筑物用多边形完全填充 (图 2)。有少数研究使用了建筑物缓冲技术 [?]，其中建筑物边界被视为一个独立的类别。这种方法可能增加模型复杂性，因此没有包括在本研究中。植被真实掩码是使用归一化植被指数 (NDVI) 生成的，二值分类的阈值通过人工检查确定。在将二值掩码映射到相应的 MX 图像后，它们被存储在 Zarr 格式中。

为了准确表示不同的土地利用类别，我们必须解决重叠问题，如建筑物与植被重叠或道路跨越水体。为了解决这些冲突，我们将建筑物、道路和水体的二值掩码组合成单一的多类别掩码。每个类被分配一个唯一的整数值，其中更具主导性的类别 (如建筑物) 被赋予较低的值。在发生重叠的情况下，选择各个掩码中的最大值，确保少数类别得到准确反映。没有标签的区域被分类为“其他”。最终，数据集被划分成 ($4 \times 256 \times 256$) 的块以生成用于训练的图像块。对于多光谱图像中超过 50% 缺失值的块，或超过 65% 像素属于“其他”类的块，都被弃用，以提高训练数据集的质量。此过滤确保了训练数据有超过 35% 的标记区域。使用这种方法，我们获得 60848 个图像块，覆盖总土地面积为 5091.94 km^2 。然而，此过滤策略并不保证训练数据 100% 被标记，如图 2 所示，也不能防止在地面真值掩码中可能出现的错误标记。

2.2 DN 到 ARD 的转换

大气顶层 (TOA) 表现反射 (ρ^*) 可以与大气底层 (BOA) 反射相关联，假设为朗伯面，如在 [?] 中所述。

其中， θ_s 、 θ_v 、 $\Delta\phi$ 和 λ 分别是太阳天顶角、观测天顶角、相对方位角和波长。 ρ 表示表现反射率，获得方式为 $\rho = \frac{\pi L}{F_0 \mu_s}$ (具体见 [?])。 ρ_s 代表 BOA (地

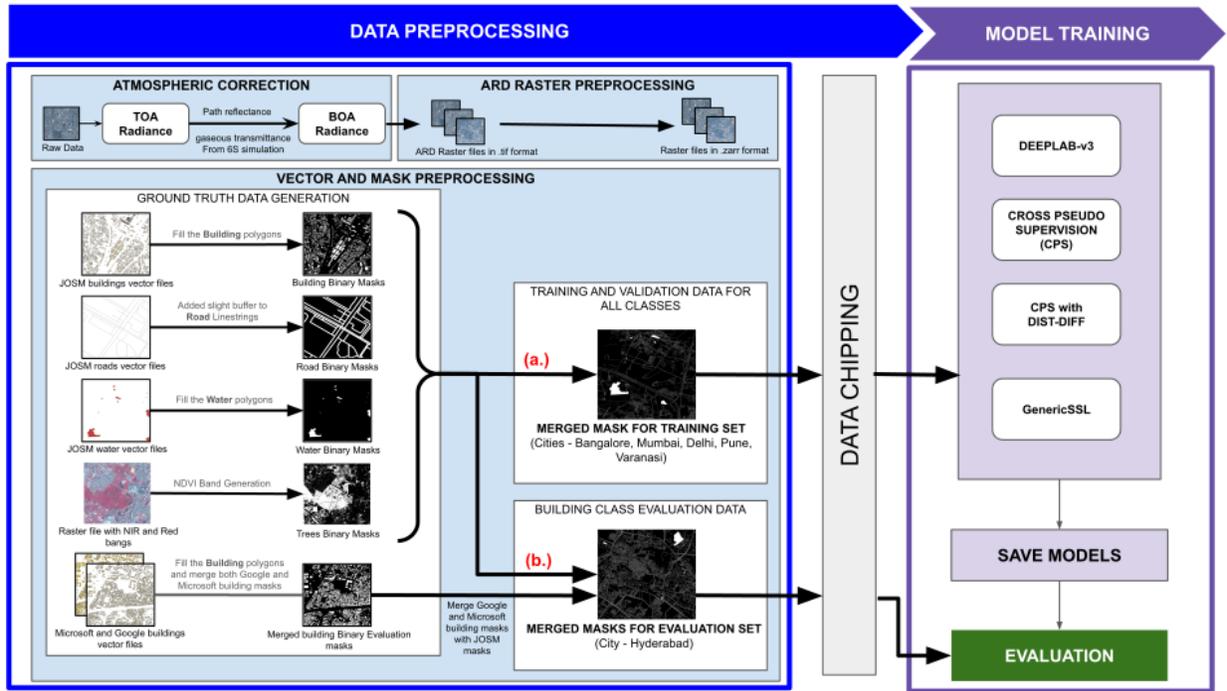


Figure 1: LULC 分割的工作流程。(a.) : 合并的训练掩码是通过结合相关城市的二进制类掩码创建的, 这些掩码是使用 JOSM 矢量文件生成的。除这些类别外, 其余区域归类为“其他”类别。在训练过程中, 我们专注于类密集分布的数据子集, 并从中提取补丁。(b.) : 合并的评估掩码是利用道路、水体、树木的输出二进制掩码、建筑物 (JOSM、Microsoft 和 Google) 的合并数据创建的, 并将其余区域视为“其他”类别。在模型评估期间, 数据被分割而不施加类大量约束。



Figure 2: 稀疏标签 (建筑物)

表) 反射率, ρ_{ra} 是由气溶胶和分子共同作用造成的路径反射率。 $T(\theta_s, \lambda)$ 和 $T(\theta_v, \lambda)$ 分别是向上和向下的大气透射率, S 是球形反照率, 而 T_g 指的是气体 (臭氧和水汽) 透射率。

6S [?] 用于模拟查找表 (LUT), 该表存储了一大

组交流系数 (a, b , 通过重排方程 ?? 中除 ρ_s 以外的未知分量获得)。系数 a 代表 ρ_{ra} , 这是气溶胶光学厚度和大气分子瑞利散射的函数, 而 b 表示气体吸收和透射, 主要是大气水汽和臭氧的函数, 主要与目标表面海拔和获取几何相关。LUT 被存储为可插值的五维对象, 以便在给定的气体浓度 (水汽和臭氧) 气溶胶光学厚度 (AOT)、地面海拔和观测几何 (θ_s, θ_v) 条件下, 插值 a 和 b 的近似值, 这些条件作为 MX 场景中每个波段的平均值获得。

我们的比较研究考虑了有监督和半监督技术。DeepLabV3+ 被用作有监督学习的基线模型架构。对于半监督方法, 我们结合了交叉伪监督, 并随后将其与两种去偏差异构模型训练方法结合: 分布-难度感知建模和难度-难度感知建模。

我们在研究中实现了基于 Segmentation Models Pytorch 库的 EfficientNet 骨干网络的 DeeplabV3+

模型作为基线。为了应对类别不平衡，该模型使用加权像素级交叉熵损失进行训练，其中类别权重设置为与类别丰度成反比。

在交叉伪监督方法中，两个具有 EfficientNet 骨干网络的 DeepLabv3+ 模型， $f(\theta_1)$ 和 $f(\theta_2)$ ，被不同地初始化并在稀疏标记的数据上进行训练，这使得可以使用相同批次的标记数据进行有监督和无监督损失计算，不同于 Xiaokang Chen 等人 (2021) [?] 的方法。监督损失包括 Hausdorff Erosion 损失 [?], L_{Sup}^{HF} ，其被类特定的权重 α_w 加权，以及加权交叉熵损失 L_{Sup}^{WCE} 。这两种损失通过基于标记图像数量和图像尺寸的常数 K 进行归一化。总的监督损失 $L_{Supervised}$ 是 L_{Sup}^{HF} 和 0.5 倍的 L_{Sup}^{WCE} 的和，如方程 3 所示。

$$K = |D_{SL}| \times W \times H \quad (1)$$

$$L_{Sup}^* = \frac{1}{K} \times \sum_{X \in D_{SL}} \sum_{w=1}^5 \sum_{i=0}^{W \times H} \alpha_w \times (l_{hd}(P_{1iw}, y_{iw}^*) + l_{hd}(P_{2iw}, y_{iw}^*)) \quad (2)$$

$$L_{Supervised} = L_{Sup}^{HF} + 0.5 * L_{Sup}^{WCE} \quad (3)$$

为了计算 CPS 中的伪监督损失，我们使用模型 2 的 softmax 输出与模型 1 的单热编码输出之间的加权交叉熵，反之亦然。这在公式 4 中详细说明，总的伪监督损失表示为 L_{CPS}^{WCE} 。

$$L_{CPS}^{WCE} = \frac{1}{K} \times \sum_{X \in D_{SL}} \sum_{w=1}^5 \sum_{i=0}^{W \times H} \alpha_w \times (l_{ce}(P_{1iw}, y_{2iw}) + l_{ce}(P_{2iw}, y_{1iw})) \quad (4)$$

在这里， P_{1iw} 和 P_{2iw} 表示来自两个模型的 logit 输出， y_{iw}^* 是真实的掩码，而 y_{1iw} 和 y_{2iw} 是分别从 P_{1iw} 和 P_{2iw} 派生的一维热向量。总损失结合了监督损失 $L_{Supervised}$ 和交叉伪监督损失 L_{CPS}^{WCE} ，如方程 5 所示。由 Laine 等人 (2016) 详述的 sigmoid 斜坡函数 [?] 随着指定时代逐渐将 L_{CPS}^{WCE} 的权重从 0 增加到 0.1，根据总的时代 T 、当前时代 t 和斜坡长度 r 来调整权衡参数 λ 。

$$L_{Total} = L_{Supervised} + \lambda \cdot L_{CPS}^{WCE} \quad (5)$$

$$\text{rampup}(r, t, T) = \begin{cases} 0 & \text{if } t = 0, \\ 0.1 \cdot \exp\left(-5.0 \left(1.0 - \frac{t}{T}\right)^2\right) & \text{if } 1 \leq t \leq r, \\ 0.1 & \text{otherwise.} \end{cases} \quad (6)$$

这里，权衡参数 λ 根据增长函数进行调整，其中 T 表示总的周期数， t 表示当前周期， r 指定增长长度。

该研究通过结合 DHC 框架中的动态加权策略对交叉伪监督 (CPS) 方法进行修改。与使用标注和未标注数据的 DHC 不同，该方法在稀疏标注数据集上运行，在相同的图像块上计算监督和无监督损失。它独立地应用了难度感知和分布感知的加权策略，并采用 sigmoid 爬坡策略逐步增加无监督损失的影响。

分布感知权重计算 (Dist): 分布感知的加权策略通过在监督和 CPS 损失函数中动态分配特定于类别的权重来解决类别不平衡问题。该方法从独热编码输出中计算每个类别的像素计数，并计算最丰富类别的像素比例与每个特定类别的像素比例，表示为 R_k (方程 7)，其中 K 为类别总数，而 k 从 1 到 K 。

$$R_k = \frac{\max\{N_k\}_{i=1}^K}{N_k} \quad (7)$$

随后，使用每个类别 k 的 R_k 与所有类别中 R_k 的最大对数值的对数比来计算权重 w_k (方程 8)。

$$w_k = \frac{\log(R_k)}{\max\{\log(R_k)\}_{i=1}^K} \quad (8)$$

初始权重是在整个训练数据集上计算的，随后对每个批次重新计算。在训练期间应用类权重的指数移动平均 (EMA)，以确保连续更新，如式 9 所示，其中 β 表示动量， W_t 代表时期 t 的权重，它们都是可调整的超参数。

$$W_t^{dist} \leftarrow \beta W_{t-1}^{dist} + (1 - \beta) W_t^{dist} \\ , W_t^{dist} = [w_1, w_2 \dots w_k] \quad (9)$$

难度感知权重计算 (Diff): 难度感知策略使用 Soft Dice 系数为每个类别计算动态权重，以评估预测结果与真实掩码之间的相似性。起初，所有类别的权重均匀分配。难度感知模型计算每个类别的 Dice 系数，并评估各个 epoch 的变化。正变化表明该类别是可学习的，而非正变化则表明该类别是不可学习的，这在

方程 10 和 11 中有定义。为每个类别调整难度后的权重计算为

$$du_{k,t} = \sum_{t-\tau}^t \mathbb{I}(\Delta \leq 0) \ln \left(\frac{\lambda_{k,t}}{\lambda_{k,t-1}} \right), \quad (10)$$

$$dl_{k,t} = \sum_{t-\tau}^t \mathbb{I}(\Delta > 0) \ln \left(\frac{\lambda_{k,t}}{\lambda_{k,t-1}} \right), \quad (11)$$

$$d_{k,t} = \frac{du_{k,t} + \varepsilon}{dl_{k,t} + \varepsilon} \quad (12)$$

$$w_k^{diff} = w_{\lambda_{k,t}} \cdot (d_{k,t})^\alpha \quad (13)$$

在这里， $du_{k,t}$ 和 $dl_{k,t}$ 分别表示未学习和已学习的度量， ε 为一个小常数，用于防止除以零。最终的类别权重经过归一化和缩放，优先考虑学习难度较大的类别，并根据学习进度进行调整。

2.2.1 通用自监督学习 (SSL)

我们实现了一个自监督学习 (SSL) 框架，用于从聚合和解耦框架 [?] 中实现的表示学习。该模型包括一个编码器，后续有三个解码器，即分布注意 (Dist)，难度注意 (Diff) 和集成 (Ens) 解码器。对 Dist 和 Diff 解码器的损失是与真实标签计算的，而集成解码器的损失则是与由其他两个解码器获得的预测集成计算的。虽然 [?] 关注使用标记数据训练两个解码器，并仅使用未标记数据训练第三个解码器，本研究使用稀疏标注的多光谱遥感图像训练所有三个解码器。

2.3 后期处理

为了减轻将大型地理空间图像划分为小块进行模型训练和预测可能导致的不准确性，我们实施了两种后期处理技术。首先，使用预测集成，将分布注意和难度注意模型的预测结果通过平均每个类别的软最大值概率结合起来。接下来，通过在评估数据集上生成滑动窗口的重叠块应用预测合并。这些块的预测结果使用最大池化方法合并回完整图像，确保预测考虑到周围的背景，从而提高准确性和整体评估分数。为了计算测试分数，使用方程 14 将软最大值概率转换为二进制输出，其中 $\rho_{i,j}$ 和 $x_{i,j}$ 分别表示 $(i,j)^{th}$ 像素的二进制值和软最大值。

$$\rho(i,j) = \begin{cases} 1, & \text{if } x_{i,j} \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

3 结果与讨论

基于查找表的 ARD 流程应用于 2023 年 3 月 12 日获取的印度海得拉巴市 MX 图像的一块补丁。校正后的图像 (图 3 b) 显示了明显的改进，有效消除了通常导致未经校正的 ρ (图 3 a) 中出现模糊外观的大气畸变。1 中列出的有监督和无监督学习模型通过计算召回率和 MIoU 得分进行评估，详见表 1，这些计算是针对代表海得拉巴 Cartosat 图像 24.6 km² 区域的补丁进行的。得分是基于使用方程 14 中的 0.4 作为阈值获得的二值化输出计算出的。图 4 描绘了使用 0.4 作为阈值而非标准 0.5 时在预测掩模中的差异。使用标准指标 (如准确率或 Dice 得分) 评估我

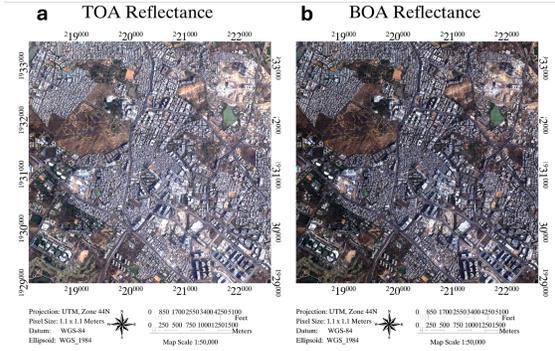


Figure 3: (a,b) 分别表示传感器获取的 TOA 反射率和 ARD 获得的 BOA 反射率

们模型的性能是具有挑战性的，因为地面实况图像的标注稀疏。这些指标不能准确评估模型的性能，因为在考虑的图像补丁方面，地面实况标签有很多错误注释的实例。相反，我们使用强调真正例和假阴性的指标。因此，我们发现召回率更适合作为我们的评估指标。表 1 显示了交叉伪监督 (CPS) 方法明显优于像 DeepLabV3+ 这样的监督模型。CPS 通过使用与类别丰富度成反比的类别权重来解决类别不平衡问题，从而提高性能。我们发现，在将动态加权策略 Dist-Diff 集成到 CPS 模型中后，类别的召回率平均有所下降，但平均交并比 (MIoU) 得分有所提高。提升了像树木这样丰富类别的召回率，同时提高了水等

Table 1: 监督和半监督模型的召回率和平均交并比分数

Model/Framework	Metric	Buildings	Water	Trees	Roads	Average
CPS with Dist-Diff	Recall	75.01	73.03	96.61	62.91	76.89
	MIoU	42.01	49.65	56.59	20.32	42.14
CPS	Recall	75.09	86.96	93.41	62.89	79.59
	MIoU	40.34	10.70	64.12	20.66	33.95
GenSSL with Ens	Recall	39.91	66.84	86.34	39.40	58.12
	MIoU	29.75	42.70	69.28	17.76	39.87
GenSSL with Ens-Diff	Recall	43.41	72.44	86.19	40.17	60.55
	MIoU	27.37	6.70	69.28	17.58	30.23
GenSSL with Ens-Dist	Recall	49.47	84.61	92.29	55.11	70.37
	MIoU	31.06	4.13	50.30	17.95	25.86
GenSSL with Ens-Dist-Diff	Recall	54.96	89.61	84.77	53.72	70.79
	MIoU	23.43	2.37	47.53	17.01	22.59
Distribution Aware (No Post Processing)	Recall	54.08	68.69	93.74	44.99	65.38
	MIoU	36.71	50.14	58.07	19.90	40.96
Difficulty Aware (No Post Processing)	Recall	38.92	63.88	79.80	36.65	54.81
	MIoU	29.73	51.36	66.84	19.27	41.80
Deeplabv3+	Recall	59.90	57.44	51.72	41.00	52.52
	MIoU	36.33	41.18	44.35	17.99	34.96

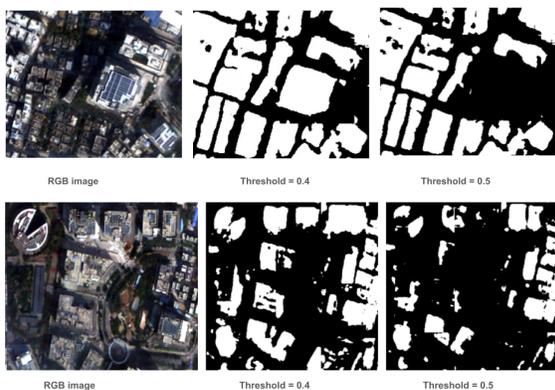


Figure 4: 图像小块的掩码阈值比较。

少数类别的 MIoU 表现。与使用低分辨率 (50 厘米) 图像并在稀疏地区进行评估的 [?] 相比, 我们的模型在高分辨率图像 (1.14 米) 上训练并在人口密集区域评估, 表现强劲。值得注意的是, 虽然通用 SSL 的得分不及 CPS, 但发现其计算代价较低, 与本研究中实现的其他半监督模型相比, 训练和部署所需的时间和空间更短, 同时仍然优于监督学习模型 Deeplab。

我们还观察到, 包括预测合并和集成在内的后处理技术改善了召回率。本案例研究使用 MX ARD 调查了海得拉巴的土地利用和土地覆盖 (LULC) 变化, 特别关注其迅速发展的 IT 中心。由于企业实体的大量土地收购和随之而来的砍伐树木, 像 Gachibowli 和 Hitech City 这些经历了广泛城市化和经济增长的地区尤其受到关注。了解这些地区的 LULC 动态对于评估 IT 部门扩张的环境影响至关重要。

该研究使用不同时间捕获的 MX 图像分析了海得拉巴的相同地区, 特别是在 2020 年 12 月和 2023 年 3 月。关注区域覆盖 24.3879 平方公里, 包括城市内的主要 IT 中心。在后处理部分描述的方法被用于预测并合并图像块回到它们原始的 NRGB 大小。合并后, 应用了模型评估过程中描述的阈值机制到 softmax 输出上。结果总结于表 1。

分析揭示了土地利用的显著变化, 特别是在 Gachibowli 和 Hitech City 的北部, 那里发生了大量的树木移除和混凝土结构的扩展 (表 2)。此外, Gachibowli 的中心湖泊已被填平, 以腾出新的建筑和道路的空间 (图 5)。我们的工作通过改进的预处理和模型评估, 提高了分类的准确性, 对 LULC (土地利用/土地覆盖) 分析作出了重大贡献。大气校正 (AC) 将原始多光谱图像转换为分析就绪数据 (ARD), 提高了最

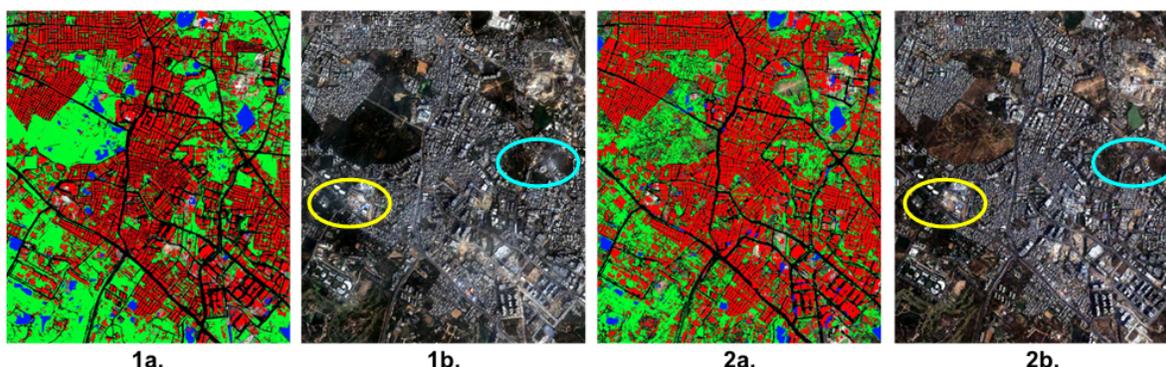


Figure 5: 索引以“1”开头的图像对应于 2020 年 12 月拍摄的海得拉巴影像，而以“2”开头的图像对应于 2023 年 3 月拍摄的影像。索引以“a”结尾的图像表示预测的掩码，索引以“b”结尾的图像表示从 CartoSAT 卫星拍摄的相应原始 NRGB 图像。在“1a.”和“2a.”中，本研究提出的模型生成的预测掩码中，红色代表建筑物，绿色代表植被，黑色代表道路，蓝色代表水体。在“1b”和“2b”中，青色圆圈标记的是 Hitech City 区域，黄色圆圈标记的是 Gachibowli 区域。

Table 2: 海得拉巴土地利用和土地覆盖的分类变化

Classes	Hyd 2020 class area (km^2)	Hyd 2023 class area (km^2)	Area Gained/Lost (km^2)	% Change
Building	8.2146	9.0611	0.8464	10.3046 %
Road	11.2745	14.8226	3.5481	31.4704 %
Water	0.8232	0.8023	-0.0209	-2.5389 %
Trees	10.1444	7.0209	-3.1235	-30.7904 %

先进的监督和半监督模型的性能。在海得拉巴市的一个区域上应用召回率 (Recall) 和 MIoU 指标进行评估，这些模型显示 CPS 在召回率方面取得了最高分，而采用动态加权 (DHC) 的 CPS 在 MIoU 方面表现卓越。召回率的提高很大程度上可以归因于 AC 过程，因为它为训练模型提供了真实的地面反射特征，从而提高了 LULC 分类的可靠性。视觉和统计评估进一步确认海得拉巴一个热点地区在三年内的显著 LULC 变化，即由于快速城市化导致的植被减少。总体而言，这项研究展示了我们集成 AC 的分割方法在捕捉土地利用变化方面的有效性，这对更精确的环境和城市规划有重大贡献。

我们在方法部分中提到的框架都是用 PyTorch 实现的，使用了 2 个 NVIDIA 4090 GPU、2 个 i9 13 代 CPU 和 256GB 的内存。我们为模型训练部分中提到的所有框架设置了 MLOps 管道（其工作流程在图 1 中简述）使用一个叫 Kedro 的开源工具。Argo 是 Katib 工人中使用的工作流程引擎。Katib 通过

Kubeflow 进行部署和编排。由于 Argo 支持有向无环图 (DAG) 和基于步骤的工作流程，它允许我们在 Argo 中运行 Kedro 管道。这使得多个管道能够并行运行，从而减少了执行时间。网络参数使用 Adam 优化器，以 0.001 的初始学习率进行优化，我们采用了一种“多项式”衰减策略，参考 [?]。由于使用的 NRGB 图像本质上是噪声的，在训练阶段按原样使用以防止过拟合。网络训练了 350 个 epoch，批量大小为 4。