

Learning Continuous Image Representation with Local Implicit Image Function

Introduction

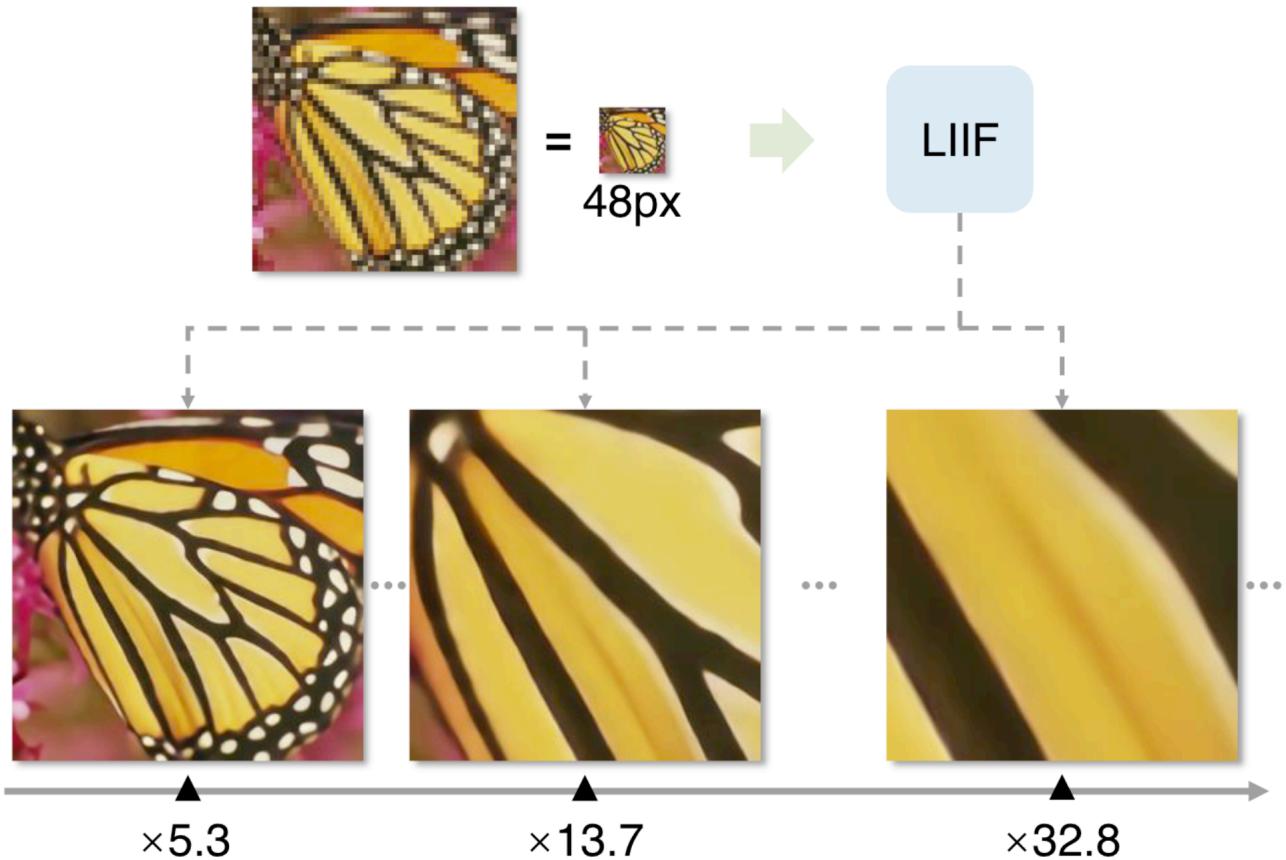


Figure 1: Local Implicit Image Function (LIIF) represents an image in continuous domain, which can be presented in arbitrary high resolution.

我们的视觉世界是连续的，但机器通常将图像存储为离散的二维像素数组。分辨率决定了图像的复杂度与精度之间的权衡。然而，这种像素表示在计算机视觉任务中存在局限性，比如当需要处理不同分辨率的图像时，通常需要统一缩放到相同大小，这可能会牺牲图像的细节和真实性。

为了解决上述问题，本文提出了一种基于连续表示的图像建模方法，即 **Local Implicit Image Function (LIIF)**。通过将图像表示为连续域上的函数，可以在任意分辨率下还原和生成图像。这种表示方式不依赖固定的分辨率，并能够在训练任务中未见过的极高分辨率（例如高达 $\times 30$ 倍）下进行图像外推。

$$f_{\theta}(z, x) = s \quad (1)$$

其中， f_{θ} 是一个多层感知机 (MLP) 表示的解码函数， z 是局部的特征向量， x 是图像中的连续坐标， s 是预测的信号值（例如 RGB 值）。

Related Work

隐式神经表示 (Implicit Neural Representation)

隐式神经表示通过多层感知机 (MLP) 将坐标映射到信号（例如，3D 形状的距离函数或图像中的 RGB 值），在 3D 对象建模领域得到了广泛应用。例如，Signed Distance Function (SDF) 可以被用来表示形状，通过以下公式描述：

$$f_{\theta}(x) = \text{SDF}(x) \quad (2)$$

其中， x 是输入的空间坐标， f_{θ} 是一个深度神经网络的参数化函数。

学习隐函数空间 (Learning Implicit Function Space)

最近的研究进一步提出共享隐式函数空间的方法，不再为每个对象单独学习一个隐函数，而是通过共享的编码器 (Encoder) 提取每个对象的潜在编码 (Latent Code)。例如，以下公式展示了这种共享方式的解码过程：

$$z = E_\phi(I) \quad (3)$$

$$f_\theta(z, x) = s \quad (4)$$

其中， E_ϕ 是编码器， z 是从输入图像 I 中提取的潜在编码， f_θ 是解码函数，用于预测信号 s （如 RGB 值）。

然而，这些方法在处理复杂自然图像时仍然存在局限性，例如无法捕获图像中的高细节特征。

图像生成与超分辨率 (Image Generation and Super-Resolution)

传统的图像超分辨率方法通常依赖固定比例的上采样。例如，基于卷积神经网络的方法通常以下方式对低分辨率输入 I_{low} 进行处理：

$$I_{\text{high}} = D(G(I_{\text{low}})) \quad (5)$$

其中， G 是特征提取网络， D 是上采样模块。然而，这些方法局限于特定比例，无法推广到任意分辨率。

与之不同，LIIF 能够在任意分辨率下进行连续超分辨率建模，其核心思想如下：

$$s = f_{\theta}(z, x) \quad (6)$$

这里的 z 表示局部特征， x 是像素的连续坐标， s 是预测信号。

现有方法的局限性

尽管上述方法在 3D 建模和简单图像任务中表现出色，但它们在处理复杂自然图像（如高分辨率照片）时往往表现不足。LIIF 提出了基于局部隐式编码的新方法，通过以下公式实现了更高的细节保留：

$$I(x_q) = \sum_{t \in \{00, 01, 10, 11\}} S_t \cdot f_{\theta}(z_t^*, x_q - v_t^*) \quad (7)$$

其中， S_t 是权重， z_t^* 是最近的潜在编码， v_t^* 是对应的坐标。

这种方法不仅能够在训练范围内表现出色，还能推广到训练任务中未见过的极高分辨率任务。

Local Implicit Image Function

本文提出了一种基于局部隐式图像函数的连续表示方法，允许图像在任意分辨率下进行渲染。以下是其核心思想和公式的详细解析。

图像的连续表示

核心思想：将图像表示为一个函数，该函数将 **图像坐标** 映射到 **信号值**（例如 RGB 值）。具体表示如下：

公式

$$s = f_{\theta}(z, x) \quad (8)$$

含义

1. s : 预测的信号值，通常是 RGB 值。
 2. f_{θ} : 解码函数，使用多层感知机 (MLP) 表示，负责将输入映射到输出。
 3. z : 局部潜在编码，表示图像中某一位置的特征向量。
 4. x : 二维坐标，定义在图像的连续域中。
-

RGB 值的预测

对于任意图像 $I^{(i)}$, 在图像坐标 x_q 处的 RGB 值可以通过查询最近邻的潜在编码预测得到。

公式

$$I^{(i)}(x_q) = f_{\theta}(z^*, x_q - v^*) \quad (9)$$

含义

1. z^* : 与坐标 x_q 距离最近的潜在编码。

- 在特征图 $M^{(i)}$ 中, 通过欧几里得距离选择最近的潜在编码:

$$z^* = \arg \min_z \|x_q - v\| \quad (10)$$

其中 v 是潜在编码在图像域中的坐标。

2. $x_q - v^*$: 偏移向量, 表示从最近潜在编码到查询坐标的偏移。

3. f_{θ} : 解码函数, 利用偏移信息和局部潜在编码预测信号值。

特征展开 (Feature Unfolding)

为了使潜在编码包含更多的局部信息, 我们对每个潜在编码进行邻域特征拼接操作。

公式

$$M'_{jk}^{(i)} = \text{Concat}(\{M_{j+l, k+m}^{(i)} \mid l, m \in \{-1, 0, 1\}\}) \quad (11)$$

含义

1. $M^{(i)}$: 初始特征图, 大小为 $H \times W \times D$ 。
 2. $M'^{(i)}$: 展开后的特征图, 每个潜在编码包含其 3×3 邻域特征。
 3. Concat: 拼接操作, 将邻域内的特征向量组合成一个新的特征向量。
-

局部集成 (Local Ensemble)

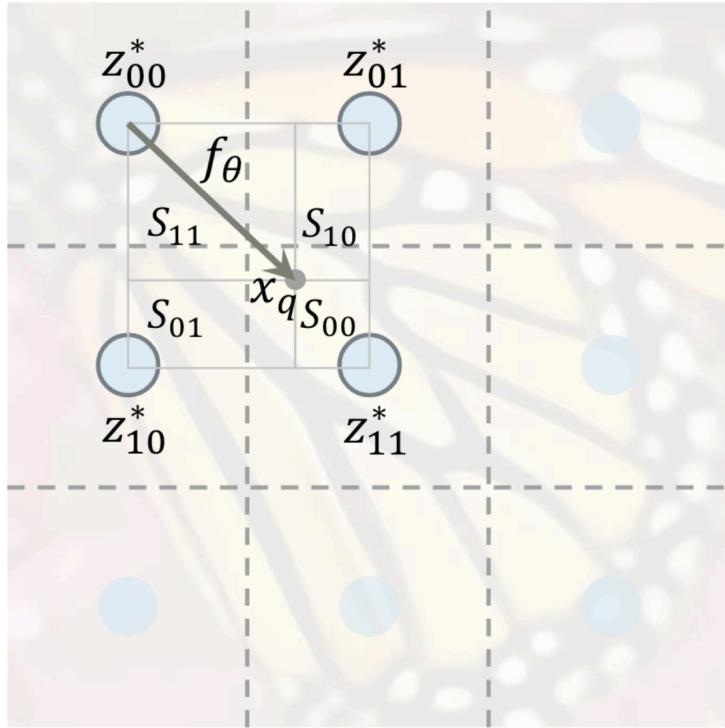


Figure 2: **LIIF representation with local ensemble.** A continuous image is represented as a 2D feature map with a decoding function f_θ shared by all the images. The signal is predicted by ensemble of the local predictions, which guarantees smooth transition between different areas.

仅使用最近邻的潜在编码可能导致预测值在区域边界处不连续。为了解决这个问题，引入了局部集成机制，通过多个潜在编码的加权预测来实现平滑过渡。

公式

$$I^{(i)}(x_q) = \sum_{t \in \{00, 01, 10, 11\}} S_t \cdot f_\theta(z_t^*, x_q - v_t^*) \quad (12)$$

含义

1. 分块：将图像域划分为左上 (00)、右上 (01)、左下 (10)、右下 (11) 四个子区域。
2. z_t^* : x_q 所在子区域的最近潜在编码。
3. v_t^* : 对应的坐标。
4. S_t : 加权系数，定义为矩形区域面积，表示 x_q 到 z_t^* 对角点的区域大小。
 - 加权公式：

$$S = \sum_t S_t \quad (13)$$

- 每个权重归一化：

$$\text{权重归一化: } S_t / S \quad (14)$$

5. 平滑性：通过四个子区域潜在编码的加权预测，确保跨区域的连续性。

单元解码 (Cell Decoding)

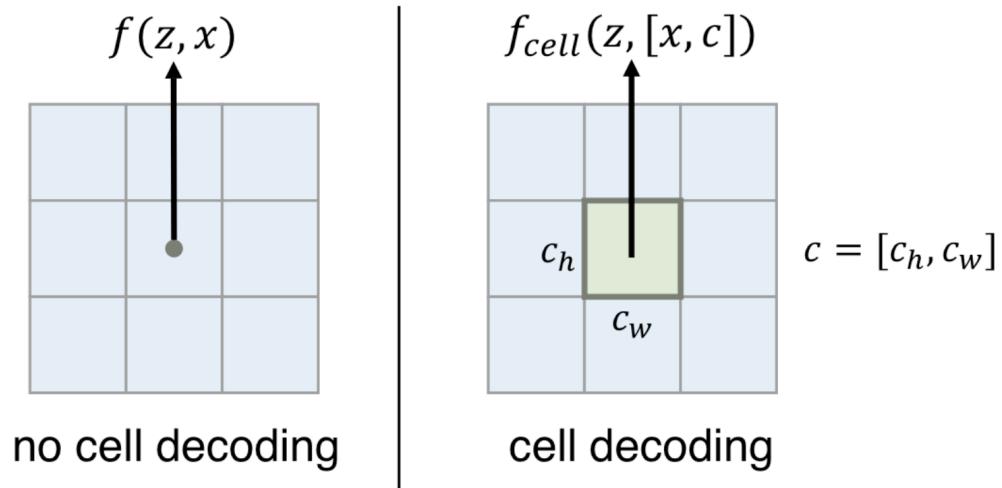


Figure 3: **Cell decoding.** With cell decoding, the decoding function takes the shape of the query pixel as an additional input and predicts the RGB value for the pixel.

为提高灵活性，LIIIF 将查询像素的形状信息作为额外输入，使预测能够适应不同的像素大小。

公式

$$s = f_{\text{cell}}(z, [x, c]) \quad (15)$$

含义

1. $c = [c_h, c_w]$: 查询像素的高度和宽度。
2. $[x, c]$: 坐标 x 与像素大小 c 的拼接。
3. f_{cell} : 扩展后的解码函数，能够根据查询像素的形状调整预测。

优势

- 查询像素的大小被纳入模型考虑，允许更精细地建模局部区域的信息。
 - 提高了模型在高分辨率图像上的适应性。
-

工作流程

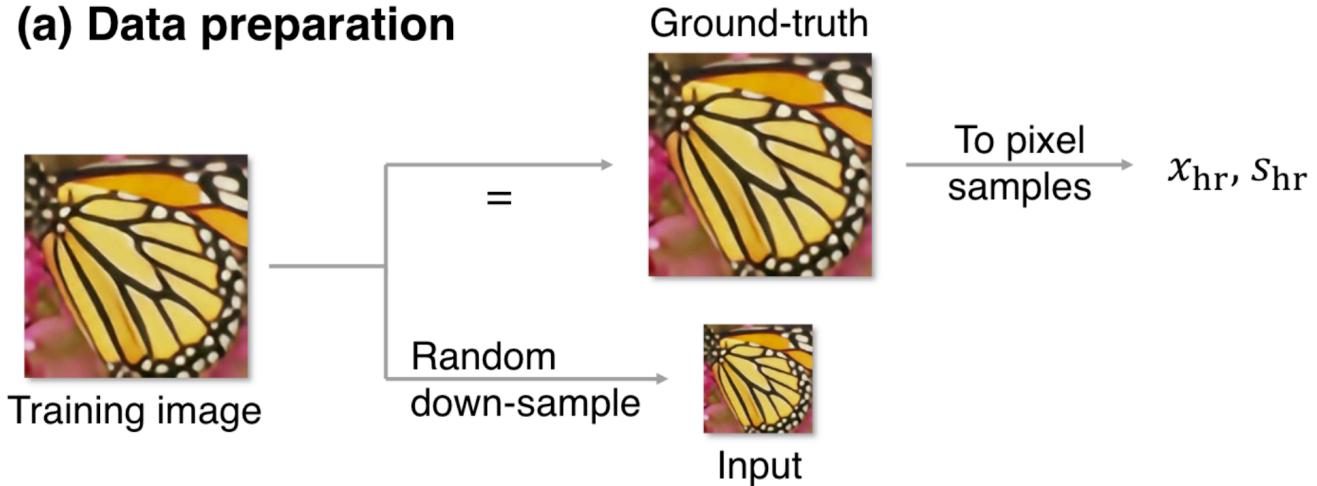
1. **输入图像**: 将离散图像 $I^{(i)}$ 转换为二维特征图 $M^{(i)}$ 。
 2. **潜在编码提取**: 在特征图中选择最近邻的潜在编码 z^* 。
 3. **解码**: 通过 $f_\theta(z^*, x_q - v^*)$ 或局部集成公式计算坐标 x_q 的信号值。
 4. **特征展开**: 对 $M^{(i)}$ 进行邻域扩展，增强局部信息。
 5. **单元解码**: 在需要时通过 f_{cell} 融合像素大小信息。
 6. **输出**: 生成连续域中的图像信号，可以在任意分辨率下渲染。
-

总结

LIIF 通过局部潜在编码和解码函数的协同工作，实现了从离散图像到连续表示的转换。这种方法支持在极高分辨率下生成图像，且在训练未涉及的分辨率上具有良好的泛化性能。

Learning Continuous Image Representation

(a) Data preparation



(b) Training

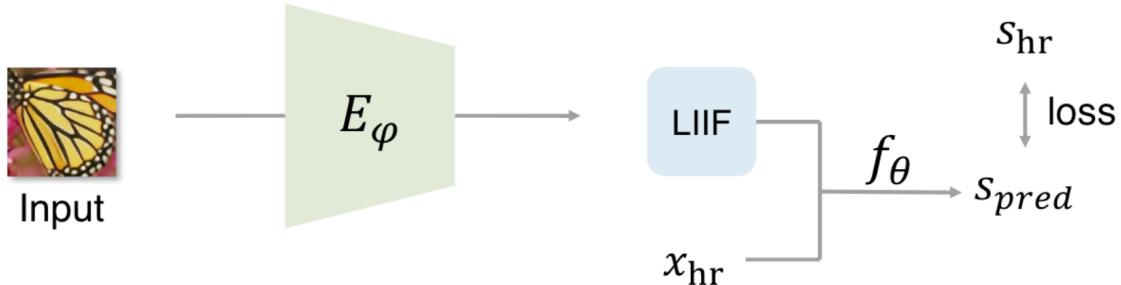


Figure 4: Learning to generate continuous representation for pixel-based images. An encoder is jointly trained with the LIIF representation in a self-supervised super-resolution task, in order to encourage the LIIF representation to maintain high fidelity in higher resolution.

本文的目标是学习从输入图像生成连续表示的方法，核心在于通过一个编码器将离散图像转化为连续的局部隐式函数表示。以下是具体方法和公式的详细解析。

方法概述

核心流程

1. **输入**：给定一组训练图像 I 。
 2. **目标**：训练一个编码器 E_ϕ 和一个解码函数 f_θ ，使得对于任意输入图像，能够生成其对应的连续表示。
 3. **任务**：通过超分辨率任务实现自监督学习，从而使生成的连续表示在更高分辨率下保持高保真。
-

数据准备

目标：从训练图像中生成连续表示所需的输入和监督信号。

数据生成流程

1. **随机降采样**：
 - 将训练图像随机缩小到不同的分辨率，生成低分辨率输入 I_{low}
 -

- 缩放因子 r 从连续范围 $[1, 4]$ 中随机采样。

2. 像素采样：

- 将高分辨率的原始图像 I_{high} 表示为像素样本对 $(x_{\text{hr}}, s_{\text{hr}})$ 。
 - 其中：
 - x_{hr} : 像素的二维坐标。
 - s_{hr} : 像素对应的 RGB 值。
-

连续表示的生成

目标：通过编码器和解码函数生成训练图像的连续表示。

编码器

编码器 E_ϕ 将输入图像 I_{low} 映射为二维特征图 M :

$$M = E_\phi(I_{\text{low}}) \quad (16)$$

其中， $M \in \mathbb{R}^{H \times W \times D}$ ， H 和 W 为特征图的大小， D 为特征维度。

解码函数

解码函数 f_θ 使用编码器输出的特征图 M ，对像素坐标 x_{hr} 进行查询，预测对应的信号值：

$$s_{\text{pred}} = f_{\theta}(z, x_{\text{hr}}) \quad (17)$$

其中：

- z 是从特征图 M 提取的局部潜在编码；
 - x_{hr} 是像素的查询坐标；
 - s_{pred} 是预测的信号值。
-

损失函数

为了训练编码器和解码函数，需要定义一个目标函数，最小化预测信号和真实信号之间的差异。

L1 损失

在实验中使用 L1 损失函数，公式如下：

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \|s_{\text{pred}}^{(i)} - s_{\text{hr}}^{(i)}\|_1 \quad (18)$$

其中：

- N 是像素样本的总数；
 - $s_{\text{pred}}^{(i)}$ 是第 i 个像素的预测信号；
 - $s_{\text{hr}}^{(i)}$ 是第 i 个像素的真实信号。
-

训练过程

数据准备

1. 从训练图像中随机采样降采样比例 r , 生成低分辨率输入 I_{low} 。
2. 从高分辨率图像中提取像素样本对 $(x_{\text{hr}}, s_{\text{hr}})$ 。

网络优化

1. 使用编码器 E_ϕ 生成特征图 M :

$$M = E_\phi(I_{\text{low}}) \quad (19)$$

2. 使用解码函数 f_θ 对像素坐标 x_{hr} 查询特征图, 生成预测信号:

$$s_{\text{pred}} = f_\theta(z, x_{\text{hr}}) \quad (20)$$

3. 计算预测信号与真实信号之间的 L1 损失:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \|s_{\text{pred}}^{(i)} - s_{\text{hr}}^{(i)}\|_1 \quad (21)$$

4. 使用梯度下降算法优化编码器和解码函数的参数。
-

连续表示的优势

通过上述流程生成的连续表示具有以下优势:

1. 任意分辨率呈现：

- 连续表示允许在任意分辨率下生成图像，而不局限于训练任务中的分辨率范围。

2. 外推能力：

- 即使在训练中未见过的高分辨率下（如 $\times 30$ 放大），连续表示仍能保持较高保真度。

3. 自监督训练：

- 使用超分辨率任务进行自监督学习，无需额外的标注数据。
-

总结

通过编码器和解码函数的联合优化，LIF 能够将离散图像表示为连续函数。这种连续表示既适用于高分辨率超分任务，也为未来其他图像生成任务提供了新的可能性。