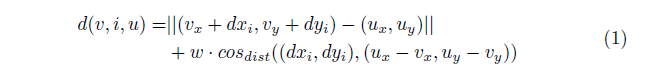


Why tensor size = W\*H\*19? Dmax = 6? 보통 도로는 최대 6거리



Cos\_dist(d\_xi, d\_yi) 방샹 측정 (stacking road detection?)

W 크게 설정 → same direction

3-1 Graph-Tensor Encoding (GTE)

For a road network graph G = {V, E}:

Region: W times H (units: meter)

GTE tensor size: W/λ \* H/λ \* (1+3 \*D\_{max})

λ : spatial resolution, 1 meter

Restricts the encoded graph, no two vertices can be co-located within a λ \* λ grid,

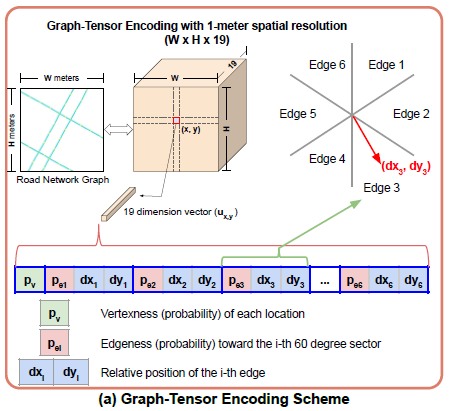
D\_{max} is the maximum edges that can be encoded at each λ \* λ grid.

 dx 和 dy 是边向量的分量，表示从一个顶点出发到其相邻顶点之间的相对位置。

 具体来说，dx 表示两个顶点之间在水平方向上的距离差，dy 表示两个顶点之间在垂直方向上的距离差。

 例如，如果顶点A在位置 (x1, y1)，顶点B在位置 (x2, y2)，那么从A到B的边向量可以表示为 (dx, dy) = (x2 - x1, y2 - y1)。

 在编码过程中，dx 和 dy 用来描述从当前顶点出发的边指向的方向和距离，以便模型能够准确地重建道路网络的拓扑结构。

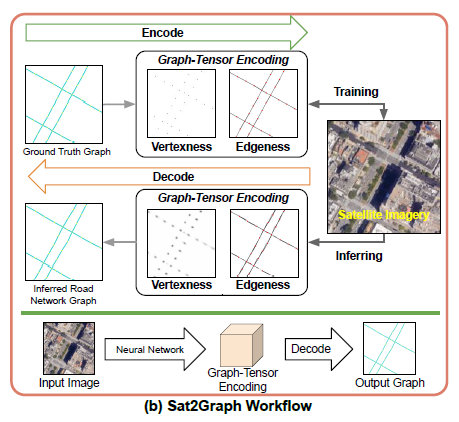


Reduce the number of possible different isomorphic encodings

Decoding: graph → tensor

使用插值，在道路上选取一些点，确定最近的两个点之间距离大于5。

P + 点连接关系

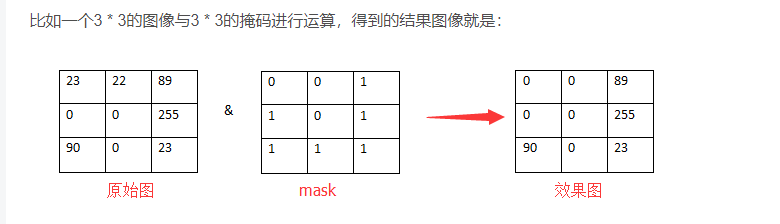


对道路段进行插值。

maximum distance threshold

road network data

The dataset is organized as 180 tiles; each tile is a 2 km by 2 km square



S2G project flow

1. Dataloader：
   1. sat\_img, gt\_seg, neighbors (dict), samplepoints dict,
2. loadtile:
   1. Read sat image(png/jpg) → normalize to ±0.45 → reshape
   2. Read neighbor data, (\_gt\_graph.p)
   3. then call neighbor\_to\_integer/neighbor\_transpose

{"complicated\_intersections": [[x1, y1], [x2, y2], ...],

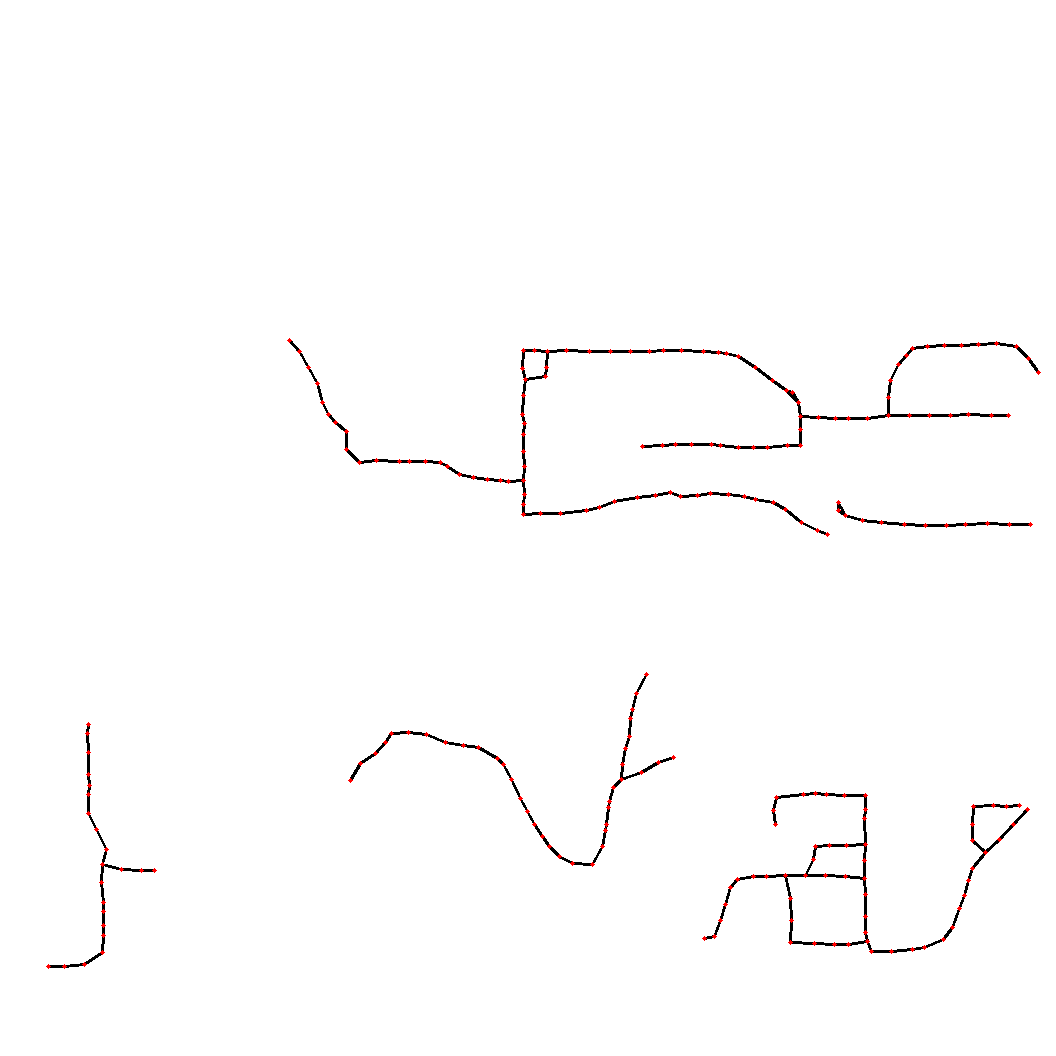
"parallel\_road": [[x3, y3], [x4, y4], ...],

"overpass": [[x5, y5], [x6, y6], ...]}

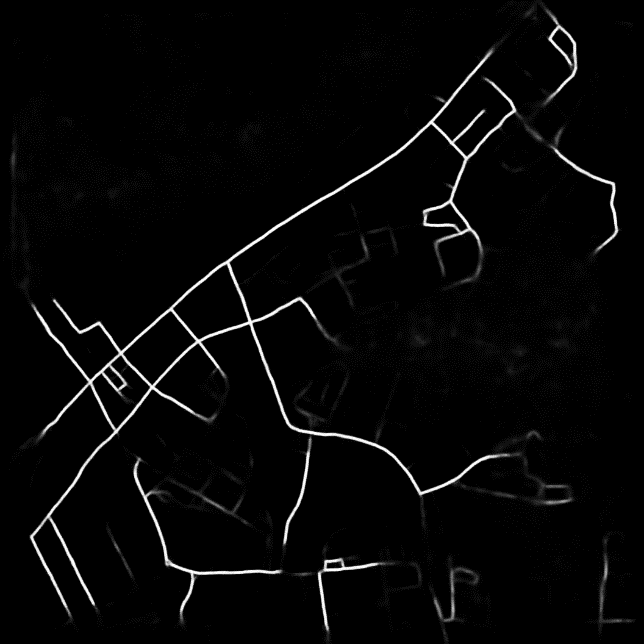
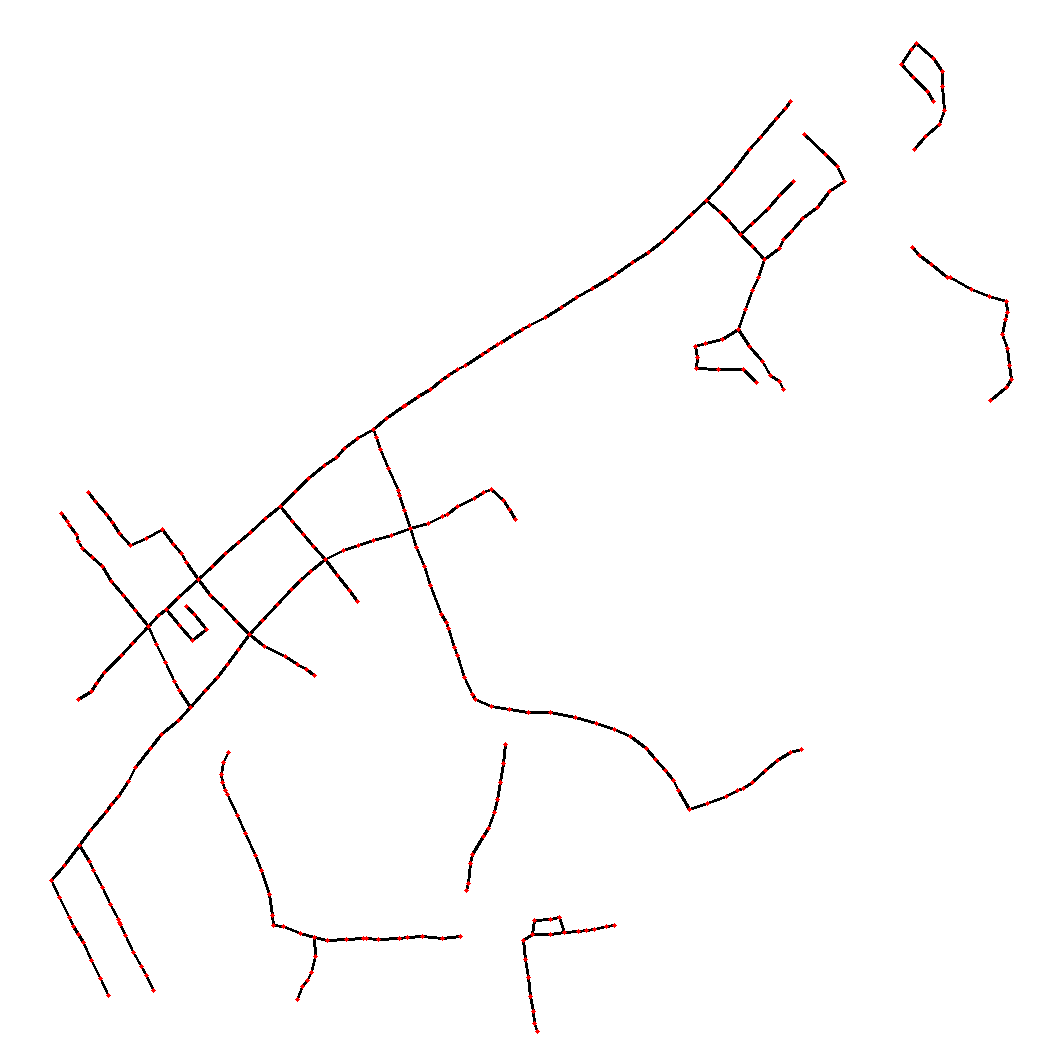
In the current implementation, we take the ground truth graph (from OpenStreetMap) as input (in graph format) and generate the corresponding segmentation mask (\_gt.png), the sample points (\_refine\_gt\_graph\_samplepoints.json), and the interpolated ground truth graphs (\_refine\_gt\_graph.p). For this part, you can check the code in prepare\_dataset/download.py

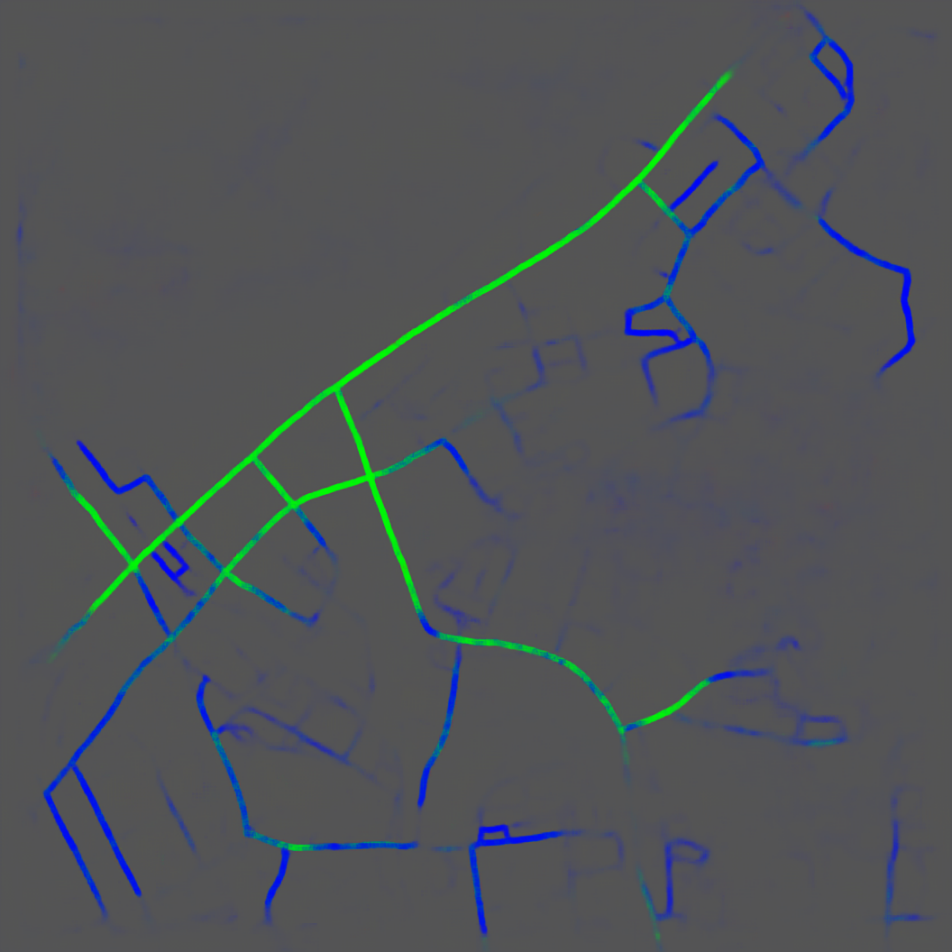
If your ground-truth is in segmentation format, then you may have to first convert it to graph format. Unfortunately, there is no code in this repo. I can try to add one if you need it.

The code to create the sample points and the refined ground truth graphs (\_refine\_gt\_graph.p).









**颜色编码总结**

* **红色 (255, 0, 0)**：表示普通节点，连接度数较低。
* **绿色 (0, 255, 0)**：表示中等重要性的节点，连接度数为2。
* **青绿色 (0, 128, 128)**：表示较重要的节点，连接度数为3。
* **蓝色 (0, 0, 255)**：表示非常重要的节点或主干道，连接度数大于等于4。



pickle模块是Python专用的持久化模块，所谓的持久化就是让数据持久化保存，可以持久化包括自定义类在内的各种数据，比较适合Python本身复杂数据的存储。但是持久化后的字符串是只能用于Python环境，不能用作与其他语言进行数据交换。pickle的本意是腌渍的意思，就是将物品永久地保存成文件，用的时候读出来还能用。

pickle模块的作用是把Python对象直接保存到文件里，而不需要先把它们转化为字符串再保存，也不需要用底层的文件访问操作，直接把它们写入一个二进制文件里。pickle模块会创建一个Python语言专用的二进制格式，不需要使用者考虑任何文件细节，它会帮你完成读写对象的操作。用pickle比打开文件、转换数据格式以及写入文件的操作能够节省不少代码。  
pickle本身和json的功能是相同的，都是将Python数据对象保存为持久化的文件，区别是**pickle能够保存Python的复杂的数据类型，包括列表、元组、自定义类等，而json只能保存字典类型的数据，同时pickle是能用Python打开，而json却可以被其他语言所读取**。

python tandong.py config/sample.json

python infer\_custom\_input.py -input region\_0\_sat.png -gsd 0.5 -model\_id 2 -output out.json

Q&A

1. BK？ Background
2. Height?
3. Student-teacher?

For your first half of the question, the answer is that we only compute the distance for one edge. In this case, even when one edge is missing, we can still connect the two vertices.

Indeed, the implementation of the decoder is complicated. I added some comments to the code and here is the outline of the decoder algorithm.

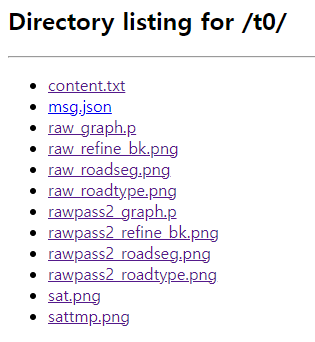
Step-1 Find vertices [(lines)](https://github.com/songtaohe/Sat2Graph/blob/7579cfcaf173db3f3c94ceb0b2f8d40878a2a204/model/decoder.py#L508-L588)

* (a) Find vertices through local minima detection [(lines)](https://github.com/songtaohe/Sat2Graph/blob/7579cfcaf173db3f3c94ceb0b2f8d40878a2a204/model/decoder.py#L509-L516) .
* (b) Find vertices at the edge endpoints. For example, we have links a<-->b and b<-->c but b is missing. In this case, we use the edges a-->b and (or) b<--c to recover b. [(lines)](https://github.com/songtaohe/Sat2Graph/blob/7579cfcaf173db3f3c94ceb0b2f8d40878a2a204/model/decoder.py#L520-L549).
* (c) Create RTree index to speed up vertex query [(lines)](https://github.com/songtaohe/Sat2Graph/blob/7579cfcaf173db3f3c94ceb0b2f8d40878a2a204/model/decoder.py#L551-L588).

Step-2 Connect the vertices to build a graph. [(lines)](https://github.com/songtaohe/Sat2Graph/blob/7579cfcaf173db3f3c94ceb0b2f8d40878a2a204/model/decoder.py#L589-L784)  
We connect vertices through three passes. Here, we use d(a-->b) to represent the distance metric for edge a-->b.

* Pass-1 For a link a<-->b, we connect them only if d(a-->b) + d(a<--b) <= snap\_dist. [(lines)](https://github.com/songtaohe/Sat2Graph/blob/7579cfcaf173db3f3c94ceb0b2f8d40878a2a204/model/decoder.py#L617-L677)
* Pass-2 (relaxed) For a link a<-->b, we connect them only if 2d(a-->b) <= snap\_dist or 2d(a<--b) <= snap\_dist. [(lines)](https://github.com/songtaohe/Sat2Graph/blob/7579cfcaf173db3f3c94ceb0b2f8d40878a2a204/model/decoder.py#L679-L708)
* Pass-3 (more relaxed) For a link a<-->b, we connect them only if d(a-->b) <= snap\_dist or d(a<--b) <= snap\_dist. [(lines)](https://github.com/songtaohe/Sat2Graph/blob/7579cfcaf173db3f3c94ceb0b2f8d40878a2a204/model/decoder.py#L710-L747)

Step-3 Some common graph post-processing passes. We apply them to all methods in our evaluation.



Heuristic 启发式的

Morphological 形态学

state-of-art 最先进的

“et al.” meaning “and others”

Cascaded 级联

backbones 骨干

one shot 一次性的

holistic 整体

trajectory 弹道

spatial resolution 空间分辨率

Interpolates 插值

Overlapping 重叠

Cosine distance

**余弦相似性**通过测量两个[向量](https://zh.wikipedia.org/wiki/%E5%90%91%E9%87%8F)的夹角的[余弦](https://zh.wikipedia.org/wiki/%E4%BD%99%E5%BC%A6)值来度量它们之间的相似性。

余弦相似性，可以被看作是在比较過程中把文件长度正規化的方法。

Tile 图块

解决方案分为两类：基于像素分割的方法和基于图的方法。基于分割的方法为卫星图像中的每个像素分配一个道路分数。然后，他们使用启发式方法提取道路网络图。在这里，道路分割充当道路网络图的中间表示。相反，基于图的方法直接从卫星图像构建道路网络图。最近，Bastani 等人 [2] 以及几项后续工作 [10,21] 利用基于图的解决方案，迭代地将顶点和边添加到部分构建的图中。我们观察到这两类方法经常互相权衡。基于分割的方法通常具有更宽的接受域，但依赖于中间非图表示和后处理启发式方法（例如，形态细化和线跟踪）从该中间表示中提取道路网络图。中间非图形表示的使用限制了基于分割的方法，因此与基于图形的方法相比，它们通常会产生噪声大且精度较低的道路网络。为了鼓励神经网络模型更多地关注道路网络的图形结构，最近的研究 [3] 提出将道路分割模型与道路方向联合训练，该方法通过这种联合训练策略实现了更好的道路连通性。然而，仍然需要后处理启发式方法。相比之下，基于图形的方法 [2, 10, 21] 直接学习图形结构。因此，与原始基于分割的方法 [2] 相比，基于图形的方法可以生成具有更好道路连通性的道路网络图。然而，图形生成过程通常是迭代的，从而导致神经网络模型更关注局部信息而不是全局信息。为了考虑更多的全局信息，最近的研究 [10, 21] 提出使用顺序生成模型来改进基于图的方法，与其他最先进的方法相比，其性能更好。

（即来自其他类别）声称具有优势。基于这一观察，一个自然而然的问题是，是否有可能将基于分割的方法和基于图的方法结合成一种统一的方法，从而从两者的优势中受益？

我们对这个问题的回答是一种新的道路网络提取方法 Sat2Graph，它将基于分割的方法和基于图的方法的固有优势结合到一个简单的统一框架中。为此，我们设计了一种新颖的编码方案，即图张量编码 (GTE)，将道路网络图编码为张量表示，从而可以训练一个简单的、非循环的、有监督的模型，该模型可以从输入图像中整体预测图结构。

除了基于张量的网络编码之外，本文还做出了两个贡献：

1. 在对 20 个美国城市 720 平方公里面积的大型城市规模数据集和流行的 SpaceNet 道路数据集 [30] 的评估中，Sat2Graph 在广泛使用的拓扑相似性度量的所有精确度-召回率权衡位置上都超越了最先进的方法。

2. Sat2Graph 可以自然地推断出堆叠道路，而之前的方法无法处理。

2 相关工作

传统方法。

从卫星图像中提取道路网络已有悠久历史 [14, 31]。传统方法通常使用启发式和概率模型从图像中推断道路网络。例如，Hinz 等人。 [20] 提出了一种通过复杂道路模型创建道路网络的方法，该模型使用有关道路和环境背景（例如附近的建筑物、车辆等）的详细知识构建而成。Wegner 等人 [32] 提出使用高阶条件随机场 (CRF) 对道路网络进行建模。他们首先将航拍图像分割成超像素，然后基于 CRF 模型连接这些超像素。

基于分割的方法。

随着深度学习的日益普及，研究人员使用卷积神经网络 (CNN) 从卫星图像中提取道路网络 [3、6、9、22、35、36]。例如，Cheng 等人 [9] 使用端到端级联 CNN 从卫星图像中提取道路分割。他们对道路分割应用二元阈值，并使用形态细化来提取道路中心线。然后，通过追踪单像素宽度的道路中心线生成道路网络图。许多其他基于分割的方法对这个基本的图形提取流程提出了不同的改进，包括改进的 CNN 主干 [6, 36]、改进的后处理策略 [22]、改进的损失函数 [22, 25]、结合 GAN [11, 28, 34] 和联合训练 [3]。

与现有的基于分割的方法相比，Sat2Graph 不依赖道路分割作为中间表示，而是直接学习图形结构。

基于图的方法。

基于图的方法直接从卫星图像构建道路网络图。最近，Bastani 等人 [2] 提出了 RoadTracer，这是一种基于图的方法，用于以迭代方式生成道路网络。该算法从路线图上的已知位置开始。然后，在每次迭代中，该算法使用深度神经网络通过查看当前位置周围的卫星图像来预测沿路要访问的下一个位置。最近的研究 [10, 21] 通过应用顺序生成模型 (RNN) 迭代生成道路网络，改进了基于图的方法。与 RoadTracer [2] 相比，顺序模型的使用允许图生成模型考虑更多的上下文信息。与现有的基于图的方法相比，Sat2Graph 一次性（整体）生成道路图。这使得 Sat2Graph 能够轻松捕获全局信息并更好地协调顶点位置。Sat2Graph 的非循环属性也使其易于训练和扩展（例如，将 Sat2Graph 与 GAN 结合起来）。我们认为 Sat2Graph 的这种简单性是优于其他解决方案的另一个优势。

使用其他数据源和其他数字地图推理任务。

从其他数据源提取道路网络也得到了广泛的研究，例如使用从移动车​​辆收集的 GPS 轨迹 [1,5,8,12,13,

18, 29]。除了道路拓扑推断之外，卫星图像还可以推断不同的地图属性，包括高清道路细节 [19,23,24]、道路安全 [26] 和道路质量 [7]。

3 Sat2Graph

在本节中，我们介绍了我们提出的方法 - Sat2Graph 的细节。

Sat2Graph 依赖于一种新颖的编码方案，可以将道路网络图编码为三维张量。我们将这种编码方案称为图张量编码 (GTE)。这种图张量编码方案使我们能够训练一个简单的、非循环的神经网络模型，将输入的卫星图像直接映射到道路网络图中（即边和顶点）。如介绍中所述，这种图构建策略结合了基于分割和基于图的方法的优点。

3.1 图张量编码 (GTE)

我们在图 2(a) 中展示了我们的图张量编码 (GTE) 方案。对于覆盖 W 米乘 H 米区域的道路网络图 G = {V, E}，GTE 使用 W\λ \* H\λ (1+3 Dmax) 3D 张量（表示为 T）来存储图的编码。这里，lambda是空间分辨率，即一米，它限制了编码图，使得没有两个顶点可以在网格内共置，Dmax 是每个网格中可以编码的最大边数。

T 的前两个维度对应于二维平面中的两个空间轴。我们在每个空间位置使用向量 ux;y = [Tx;y;1; Tx;y;2; :::; Tx;y;(1+3 Dmax)]T 来编码图形信息。如图 2(a) 所示，向量 ux;y 有 (1 + 3 Dmax) 个元素。其第一个元素 pv 2 [0; 1]（顶点度）编码在位置 (x; y) 处有顶点的概率。第一个元素之后是 Dmax 个 3 元素组，每个组都编码从位置 (x; y) 发出的潜在出边的信息。对于第 i 个 3 元素组，其第一个元素 pei 2 [0; 1]（边缘度）编码朝向 (dxi; dyi) 发出的出边的概率，即从 (x; y) 指向 (x + dxi; y + dyi) 的边。

在这里，我们将 Dmax 设置为 6，因为我们发现在道路网络图中度大于 6 的顶点非常罕见。

为了减少同一输入图可能的不同同构编码的数量，GTE 仅使用第 i 个 3 元素组对指向从 (i ? 1) 360

Dmax 度到 i 360 Dmax 度的 360Dmax度扇区的边进行编码。

我们在图 2(a) 中显示了此限制和一个示例边（红色）。此策略对编码图施加了新的限制 {对于编码图中的每个顶点，每个 360 Dmax 度扇区最多只能有一个传出边。但是，我们发现此限制不会影响 GTE 对大多数道路图的表示能力。这是因为 GTE 编码的图是无向的。我们将在第 5 节中讨论这个问题。

编码

将道路网络图编码为 GTE 很简单。对于道路网络提取应用，编码算法首先在道路网络图中插入直线路段。它选择最小数量的均匀分布的中间点，使得连续点之间的距离在 d 米以下。这种插值策略调节 GTE 中边向量的长度，使训练过程稳定。这里，d 值较小， 例如，d < 5 会将 GTE 转换回道路分段，使 GTE 无法表示堆叠道路。非常大的 d 值（例如 d = 50）会使 GTE 难以近似弯曲道路。出于这些原因，我们认为 15 到 25 之间的 d 值效果最佳。在我们的设置中，我们将 d 设置为 20。对于堆叠道路，插值可能会在同一位置产生属于两个重叠道路段的顶点。当发生这种情况时，我们使用迭代冲突解决算法来移动两个边的端点顶点的位置。目标是确保任何两个顶点之间的距离（来自两个重叠边）大于 5 米。在训练期间，这种冲突解决预处理还会为堆叠道路产生更一致的监督信号 - 重叠边往往总是在每个边的中间附近交叉。在此步骤之后，编码算法按照图 2(a) 所示的方案将每个顶点映射到 3D 张量 T。例如，当位置 (x; y) 处有顶点时，算法将 ux;y 的顶点度 (pv) 设置为 1，否则顶点度设置为 0。

解码

GTE 的解码算法将图的预测 GTE（通常有噪声）转换回常规图格式 (G = fV,Eg)。解码算法包括两个步骤，(1) 顶点提取和 (2) 边连接。由于顶点度预测和边度预测都是 0 和 1 之间的实数，我们只考虑概率大于阈值（表示为 pthr）的顶点和边。

在顶点提取步骤中，解码算法通过定位顶点图的局部最大值来提取潜在顶点（我们在图 2(b) 中展示了一个示例）。该算法仅考虑顶点大于 pthr 的局部最大值。在边连接步骤中，对于每个候选顶点 v 2 V ，解码算法将其传出边连接到其他顶点。对于顶点 v 2 V 的第 i 条边，该算法通过以下距离函数计算其与所有附近顶点 u 的距离，

首先，道路网络图是无向的。虽然道路有方向，但道路方向可以在道路网络提取后作为道路属性添加。在这种情况下，对于每个边 e = (va; vb)，我们只需要编码从 va 到 vb 或从 vb 到 va 的一个链接，而不是编码这两个链接。尽管 GTE 对来自一个顶点的出边具有 360

Dmax 度扇区限制，但这种无向图属性使得可以对非常尖锐的分支进行编码，例如高速公路和出口匝道之间的分支顶点。

其次，道路网络图是可插值的。可能存在这样的情况：边 e = (va; vb) 的两个链接都无法编码​​到 GTE 中，因为 va 和 vb 都需要对它们的其他出边进行编码。但是，由于道路网络图是可插值的，我们始终可以将边 e 插值为两个边 e1 = (va; v0) 和 e2 = (v0; vb)。插值后，原始几何和拓扑保持不变，但我们可以使用附加顶点 v0 来编码 va 和 vb 之间的连通性。

在表 3 中，我们展示了需要使用数据集中具有不同 Dmax 值的无向和可插值属性来固定的边的比例。

对监督学习的担忧。GTE 的另一个担忧是，对于一个输入图，存在许多不同的同构编码（例如，在一条长路段上有许多可能的顶点插值）。这些同构编码会产生不一致的地面实况标签。在训练过程中，地面实况的这种不一致性可能使得通过监督学习学习正确的映射变得非常困难。然而，与直觉相反，我们发现 Sat2Graph 能够通过监督学习进行学习，而且学得很好。我们发现，其关键原因是 GTE 表示的不一致性对图中的顶点和边的影响并不相同。例如，在不同的同构 GTE 中，交叉口顶点的位置始终保持一致。我们发现 GTE 在交叉口、立交桥/地下通道等重要地点的监督学习中具有较高的标签一致性。通常，这些地方是挑战真正来自的地方。虽然 GTE 对于长路段的一致性较低，但长路段的拓扑非常简单，仍然可以通过 GTE 的解码算法进行纠正。