轻量级无监督回环检测系统

概要：构造了一个轻量级的CNN模型,来生成类似HOG的描述子。用该描述子构成词典，进行回环。

开源代码：https://github.com/rpng/calc

我们在使用代码时，既可以自己重新训练模型，也可以直接使用他已经训练的模型。

代码分为两个部分：
１．TrainAndTest用于训练和测试模型，python;
２．DeepLCD用于回环检测；c++；
里面有一个demo，用的KITTI双目拍的图片，左图放入数据库，用右图搜索，就可以找到左图。

系统框架

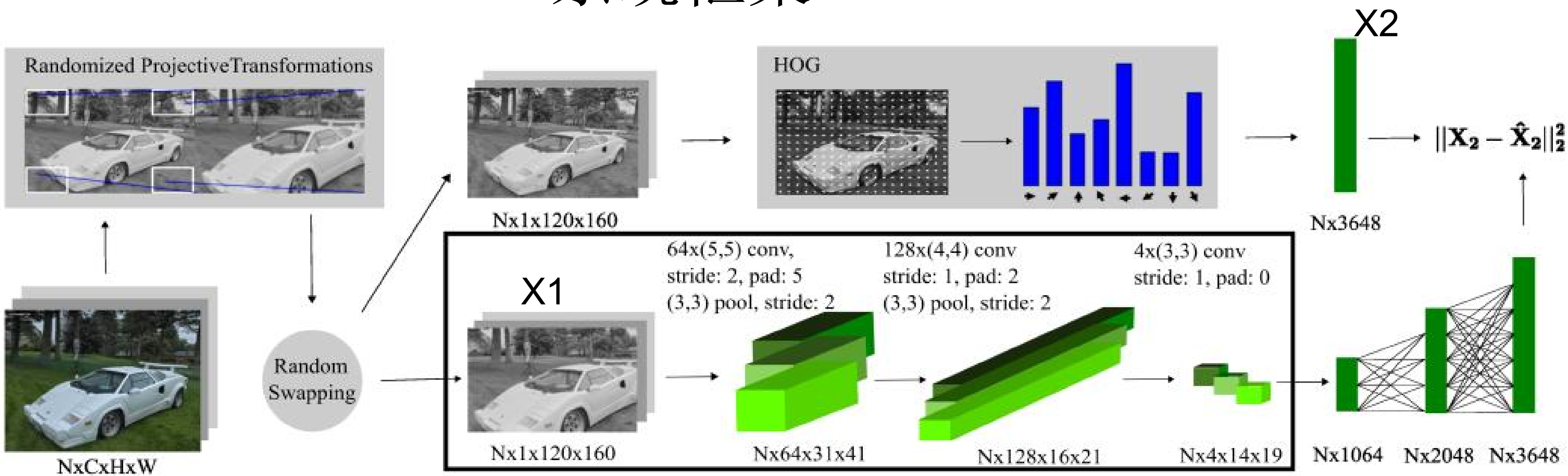


Fig. 2: The training pipeline for our deep model. In this architecture, the projective transformations and HOG descriptors are computed only once for the entire training dataset, and the results are then written to a database to use in training. Upon deployment, the batch size N is set to 1, and only the layers in the boxed area are used.

每张图片生成两张图片，组成一个图片对。

假设有N对图片对，从每一对图片对里随机选一张出来计算HOG描述子(维度3648)，把所有的HOG描述子放在一起，得到N×3648，为X2。

图片对的另一个图片就放在X1里面，得到N×120×160，为X1

系统框架

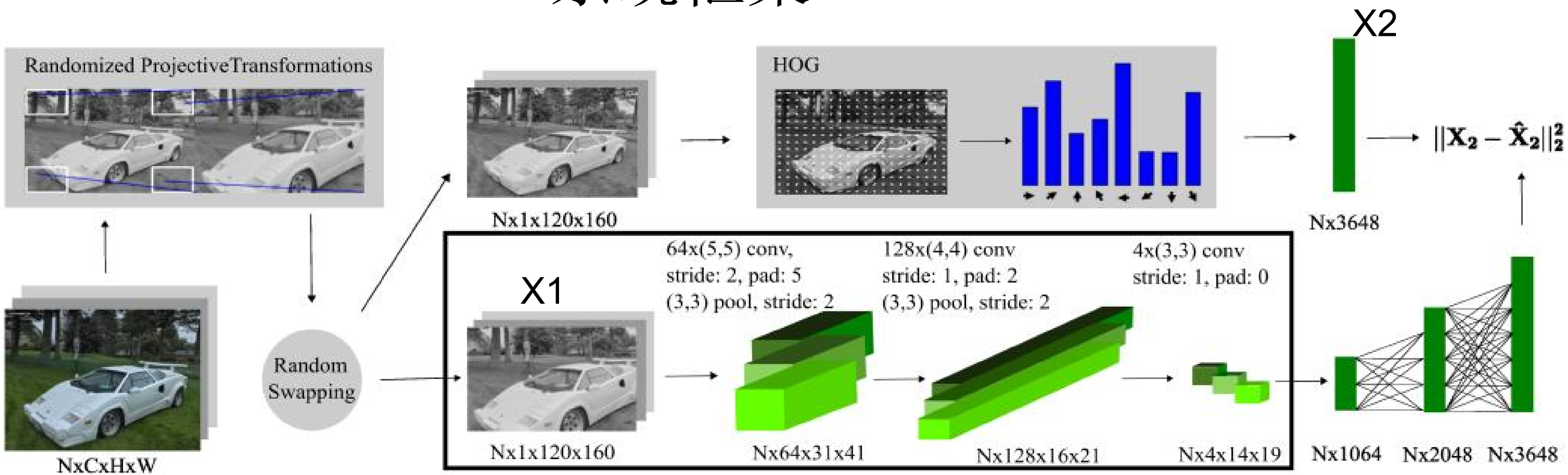


Fig. 2: The training pipeline for our deep model. In this architecture, the projective transformations and HOG descriptors are computed only once for the entire training dataset, and the results are then written to a database to use in training. Upon deployment, the batch size N is set to 1, and only the layers in the boxed area are used.

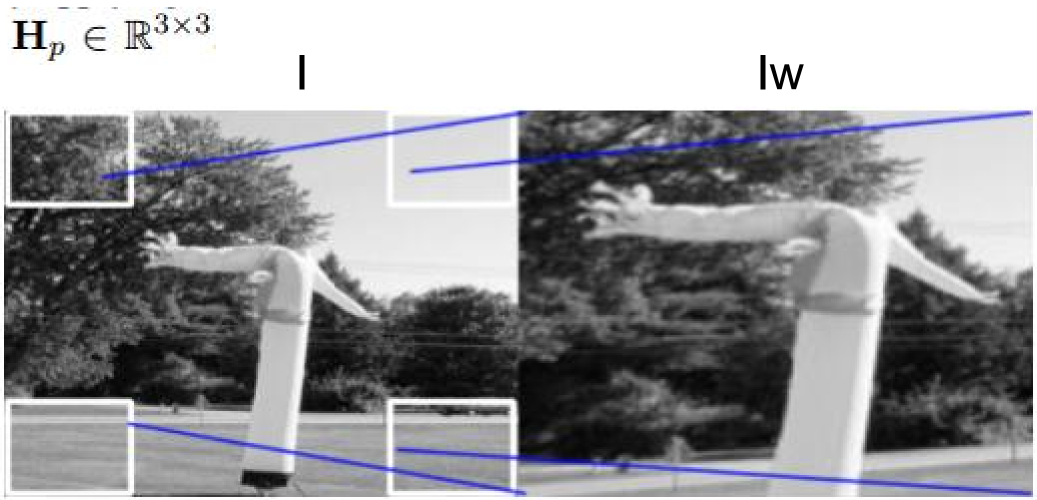
我们的模型就是用X1重构X2。　　　模型建立完成后，去掉后面3层

用两层convolution and pooling paired layers，一层纯卷积层，和三层fully-connected layers。每一层有一个激活函数。

3 个卷积层都使用修正线性单元作为激活函数， 3 个fully-connected layers都使用s型函数作为激活函数。

模型训练

一张图片生成图片对



四个框为H/4×W/4，这样既可以避免Iw过度畸变，又可以足够变形来形成该图片一个新的表达。（是不是说大了容易畸变，小了和原图差不多）

在原图(左图)的四角取四个框，在框里面分别随机取一个点，把它作为新图(右图)的四个角点。Hp就是把左图转成右图的转换矩阵。

得到Hp后，用Hp把原图I转为新图Iw，I和Iw构成图片对。

模型训练

用数据集生成X1,X2。（这里他又写成了T1 T2）

Algorithm 1 Generating Training Data

input: $\mathcal{I}$: A set of grayscale training images, resized to $H \times W$
output: $\mathbf{T}_1 \in \mathbb{R}^{M \times H \times W}$ and $\mathbf{T}_2 \in \mathbb{R}^{M \times D}$
define: $rand(\mathcal{A})$ as a map from set $\mathcal{A}$ to one of its elements, chosen at random

$W, H, D, M \leftarrow 160, 120, 3648, |\mathcal{I}|$
$\mathbf{T}_1 \leftarrow \mathbf{0}_{M \times H \times W}$
$\mathbf{T}_2 \leftarrow \mathbf{0}_{M \times D}$
$\mathbf{P}_c \leftarrow ((0,0),(0,H),(W,0),(W,H))$
for $i \in \mathbb{N} \cap [1, M]$ **do**
  $\mathbf{I} \leftarrow rand(\mathcal{I})$
  $\mathbf{P}_r \leftarrow randFourPts(W,H)$
  $\mathbf{H}_p \leftarrow estimateHomography(\mathbf{P}_r, \mathbf{P}_c)$
  $\mathbf{I}_w \leftarrow transform(\mathbf{I}, \mathbf{H}_p)$
  if $rand(\{0,1\})$ **then**
    $swap(\mathbf{I}, \mathbf{I}_w)$
  $\mathbf{T}_1^{(i)} \leftarrow \mathbf{I}$
  $\mathbf{T}_2^{(i)} \leftarrow calcHOG(\mathbf{I}_w)$

算法1描述了如何把一张图片生成图片对，并分别放入T 1和T2。

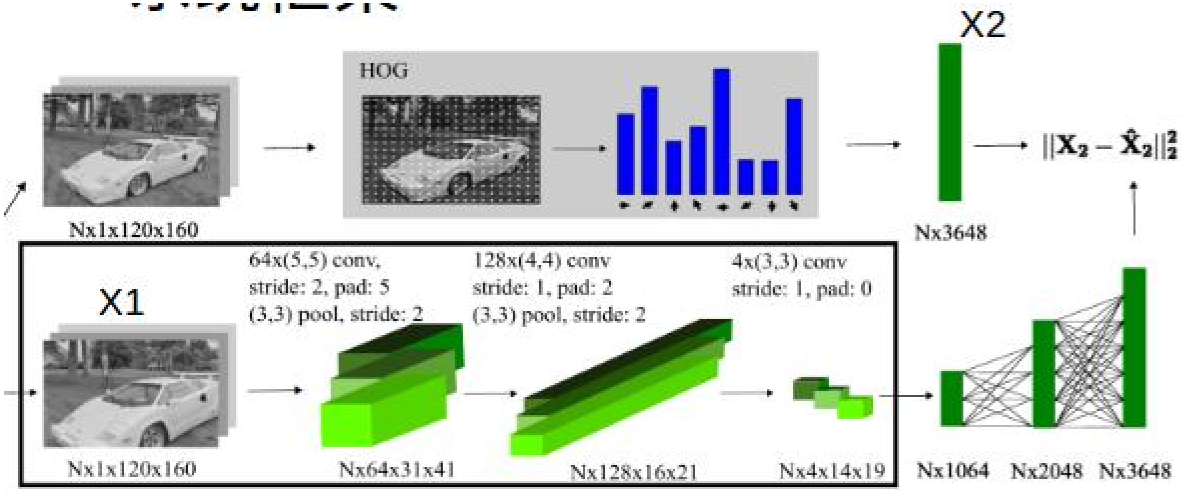
图片对里有一张是原始图片，有一张是原图通过Hp矩阵转换而来。

图片对两张图片放入T1 T2是随机的，如果一张放入T1，另外一张就计算HOG描述子，放入T2。

Iw相对于I都放大了，为了避免不必要的训练偏差，所以I和Iw是随机生成T1,T2的。

模型训练

得到X1和X2之后，就开始训练了



使用caffe框架

To construct and train our model, we utilize the Caffe Deep Learning Library [38] due to its efficiency. We train our model for roughly 42 epochs with a fixed learning rate of 9×10^{-4}. Based on Krizhevsky et al. [39], we choose a momentum of 0.9, and weight decay of 5×10^{-4}.

实际就是构造了一套生成描述子的方法

在线使用

用我们模型生成的描述子创建一个数据库，通过查询它找到回环候选帧(和DBOW2一样)，尽管K-D树是创建这类数据库的常用方法，但是搜索1064向量，它相对于线性搜索并没有速度提升。所以我们使用简单的线性搜索。另外，因为描述子足够紧凑，可以不需要降维直接计算。

Once our model is trained, upon its deployment for online use, we create a database of the descriptors extracted by our model and later query it to find loop closure candidates. While K-D trees [40] are a popular means to create such databases for nearest-neighbors searches, there is no speed up over a linear search for 1,064-dimensional vectors – even when the search is approximated [41]. For this reason, we use the simple linear

他在结论部分说明，如果想进一步优化，也可以改为K-D tree

实验结果

在不同数据集下，对比准确率和召回率

1． The Alderley Dataset

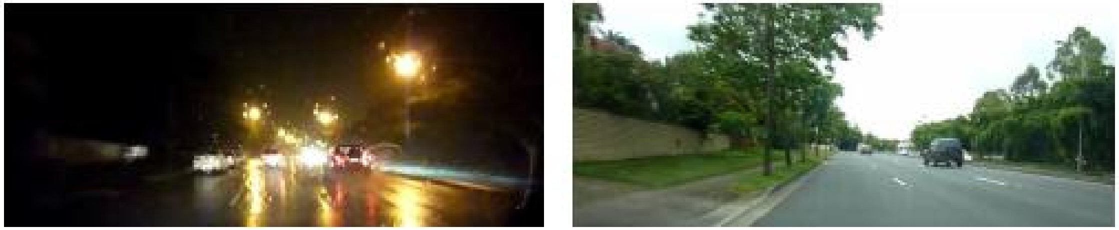


Fig. 4: An example image pair from the Alderley dataset. Note that these frames are extremely difficult to match, even for a human.

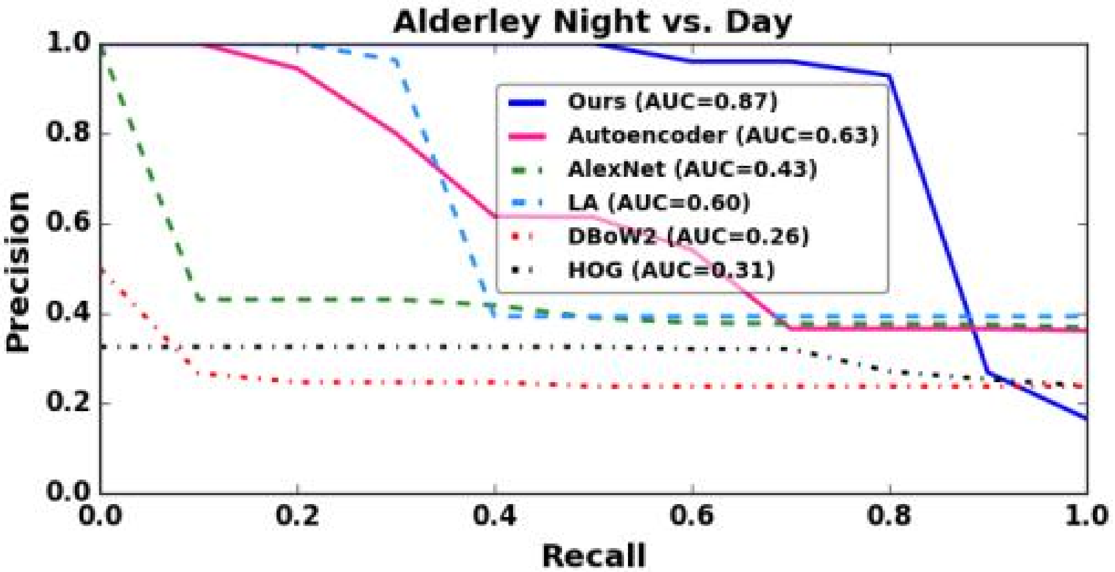


Fig. 5: Our method outperforms the state-of-the-art algorithms on the Alderley dataset, with the highest AUC and r value.

实验结果

在不同数据集下，对比准确率和召回率

2． The Alderley Dataset

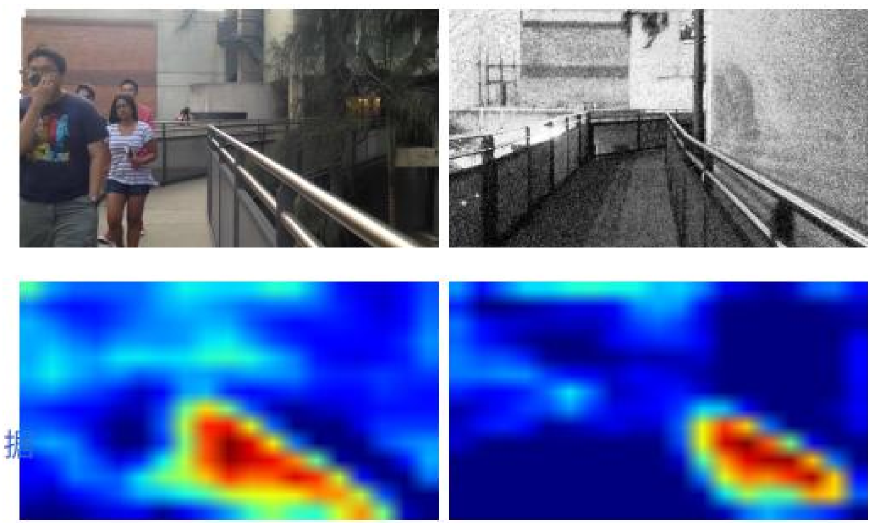


Fig. 1: An example image match from the *Gardens Point* dataset, which demonstrates large differences in viewpoint, dynamic objects, and illumination, as well as occlusions. Nevertheless, with the right image as the query, our proposed method correctly retrieves the left during our experiments (see Section IV), while all of the tested state-of-the-art methods retrieve incorrect images. Below each image, the first face of the descriptor layer, before flattening, is shown. Evidently, these visually dissimilar images are transformed into very similar activation maps.

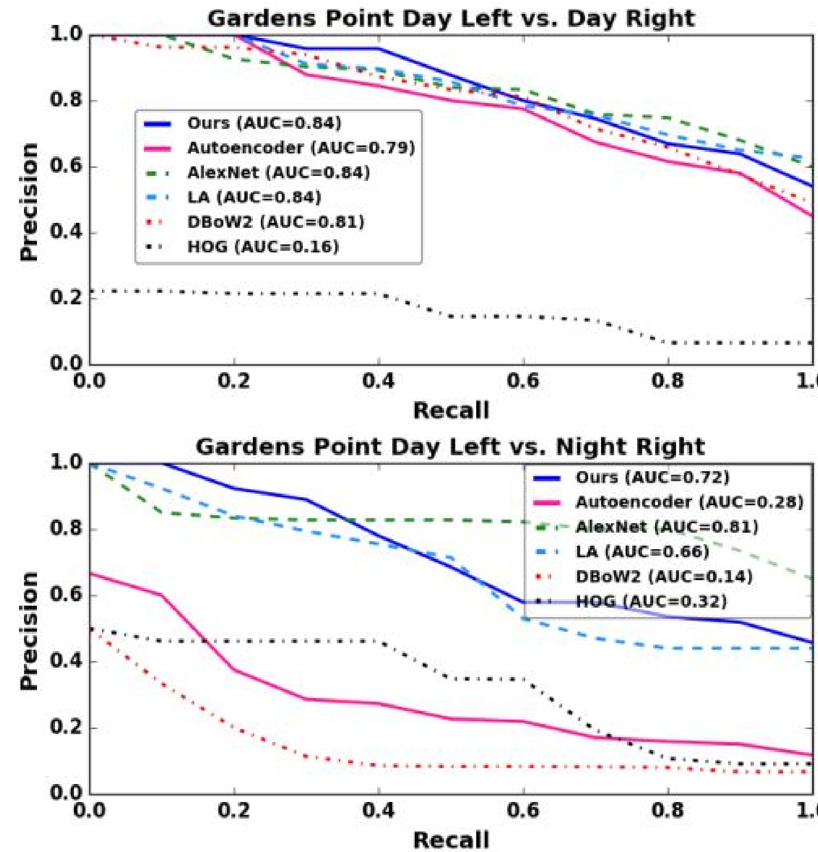


Fig. 6: The comparison results on the Gardens Point dataset. (top) Our method performs comparably with [16] (which, however, is a supervised learning approach) in the *day-time sequence*, while (bottom) our method outperforms its competitors in the *night-time sequence*.

实验结果

在不同数据集下，对比准确率和召回率

3． The Nordland Dataset

Fig. 7: An example image pair from the Nordland dataset. The left image is from the spring sequence while the right one is from the winter.

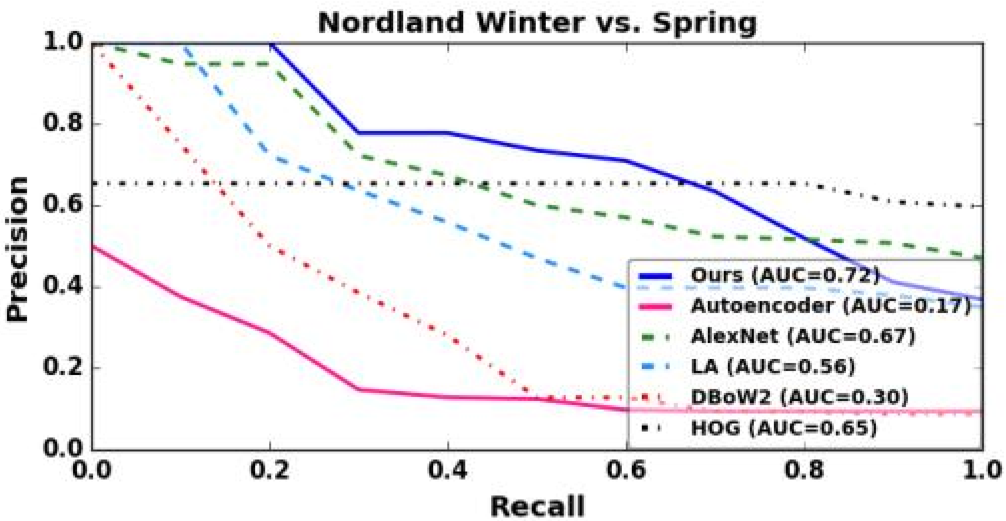


Fig. 8: Comparison results on the Nordland dataset. Our method is observed to be more robust to the seasonal changes provided by this subset of the winter and spring sequences.

实验结果

在不同数据集下，对比准确率和召回率

4． Our Campus Loop Dataset

Fig. 9: An image pair example from our Campus Loop dataset, which has extreme variations in viewpoint, weather, and dynamic objects.

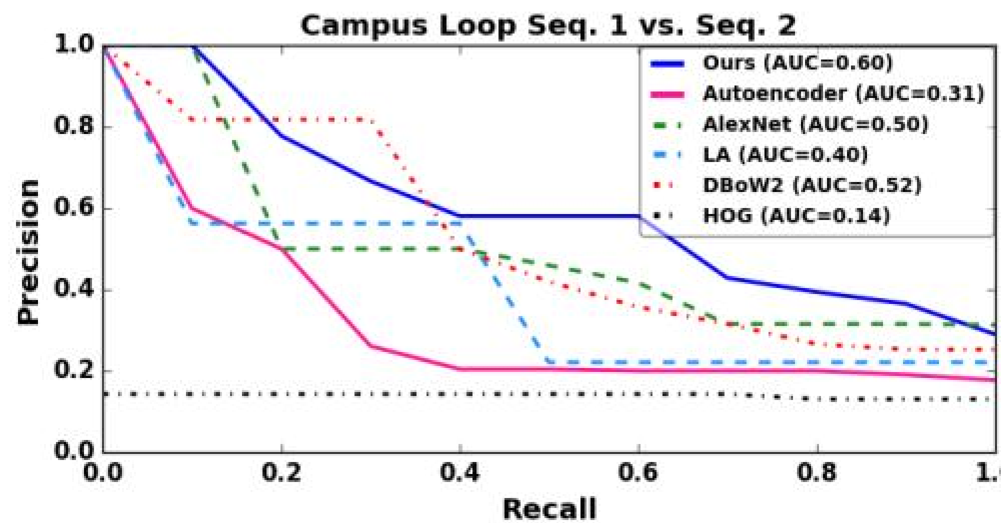


Fig. 10: Our approach outperforms the other benchmark methods on our own Campus Loop dataset, with the highest r value while tying with AlexNet `conv3` for the highest AUC.

实验结果

检测实时性

硬件平台： an i7-6700HQ CPU, anda GeForce GTX 960M GPU

TABLE I: Times (in milliseconds) to extract features and query a database of 4,541 images on the KITTI dataset.

Method	Extract (GPU)		Extract (CPU)		Query	
	μ	σ	μ	σ	μ	σ
Ours	**0.862**	0.025	44.0	2.98	**1.47**	0.031
DBoW2	N/A	N/A	15.8	3.08	4.25	0.547
AlexNet (no GRP)	2.13	0.038	405.0	17.4	80.8	0.708
AlexNet	16.6	0.658	418.0	17.8	N/A	N/A

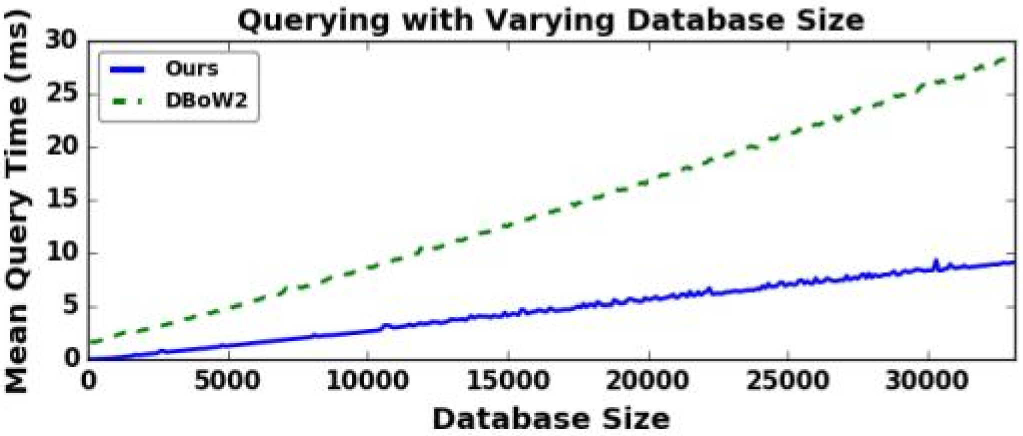


Fig. 11: The proposed method performs queries faster than DBoW2 with varying database size.

实验结果

检测在线回环

使用数据集KITTI sequences 00 and 05

回环策略： １．根据准确率和召回率选定一个得分阈值，大于该阈值认定为回环候选帧。
２．如果当前帧跟回环候选帧最近的6帧里面，有３次连续的回环，即认为回环了
３．数据库数据量必须足够且邻近帧不参与查询

seventh frame for loop detection. A loop closure hypothesis is proposed if a database query score is above an a-priori threshold τ, and a loop is determined closed if three consecutive queries retrieve descriptors within six frames of the first query. We exclude the most recent images from the search space, and do not start loop detection until the database is of sufficient size. We choose τ from the precisions and recalls shown in Fig. 6 (bottom) such that it maximizes the recall rate with

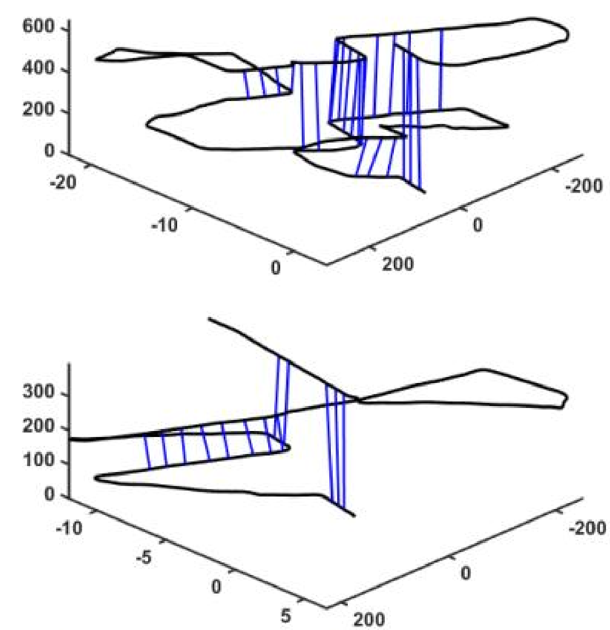


Fig. 12: The results of online loop closure using KITTI 00 and 05, respectively. The 2D location of the trajectory is represented on the x-y plane, and the z-axis is the current keyframe number.