LIFT: Learned Invariant Feature Transform

Kwang Moo Yi∗,1, Eduard Trulls∗,1, Vincent Lepetit2, Pascal Fua1

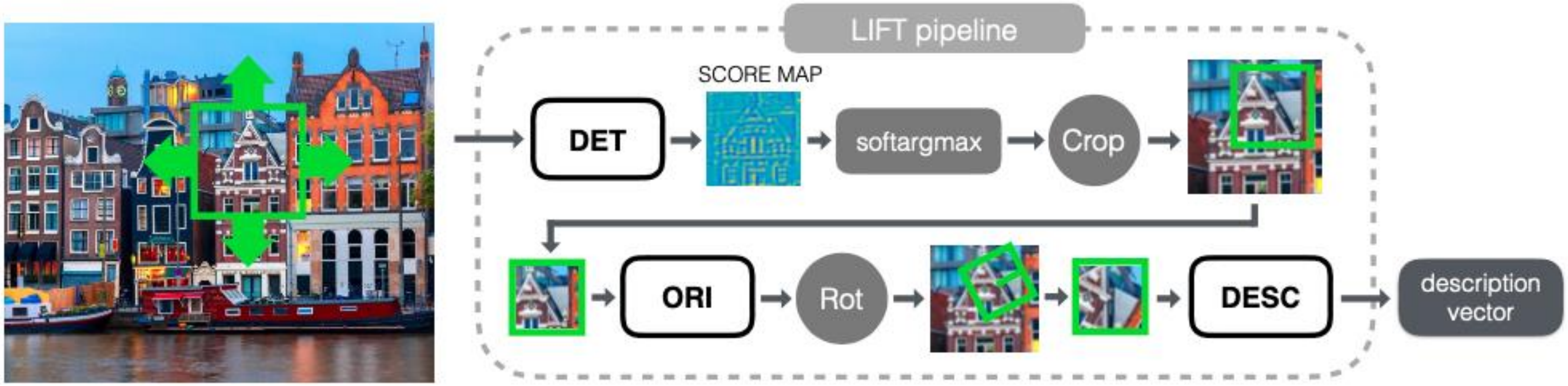
Input

- P_1: anchor point
- P_2: matching point
- P_3: non-matching point
- P_4: non-interesting point

in patch format

Whole pipeline

整个pipeline:
Detection -> orientation estimation -> description



本文有趣的点在于:

1. 整个pipeline的训练顺序: 和predict 恰好相反
2. 在中间输出没有监督的情况下, 如何通过pipeline后面的网络给前面网络提供监督.(隐式)

Descriptor

$$\mathbf{d} = h_\rho(\mathbf{p}_\theta)$$

输入: 利用SFM模型中提供的点的位置和方向来得到旋转之后的patches
输出: patch对应的描述子:
达到的目的: discrimination

$$\mathcal{L}_{\mathrm{desc}}(\mathbf{p}_\theta^k, \mathbf{p}_\theta^l) = \begin{cases} \left\| h_\rho(\mathbf{p}_\theta^k) - h_\rho(\mathbf{p}_\theta^l) \right\|_2 & \text{for positive pairs,} \\ \max\left(0, C - \left\| h_\rho(\mathbf{p}_\theta^k) - h_\rho(\mathbf{p}_\theta^l) \right\|_2\right) & \text{for negative pairs} \end{cases}$$

Orientation Estimator

$$\theta = g_{\phi}(\mathbf{p}),$$

输入: cropped patch
输出: 所谓的“方向”

$$\mathcal{L}_{\text{orientation}}(\mathbf{P}^1, \mathbf{x}^1, \mathbf{P}^2, \mathbf{x}^2) = \left\| h_{\rho}(G(\mathbf{P}^1, \mathbf{x}^1)) - h_{\rho}(G(\mathbf{P}^2, \mathbf{x}^2)) \right\|_2$$

不是通常意义上的方向, 是一个为了让后面的描述子网络work的方向, 根本原因是CNN本身不具备旋转不变性.

detector

- input: raw patch
- Output: heatmap -> mass location x

$$\text{softargmax}\left(\mathbf{S}\right) = \frac{\sum_{\mathbf{y}} \exp(\beta \mathbf{S}(\mathbf{y}))\mathbf{y}}{\sum_{\mathbf{y}} \exp(\beta \mathbf{S}(\mathbf{y}))}$$

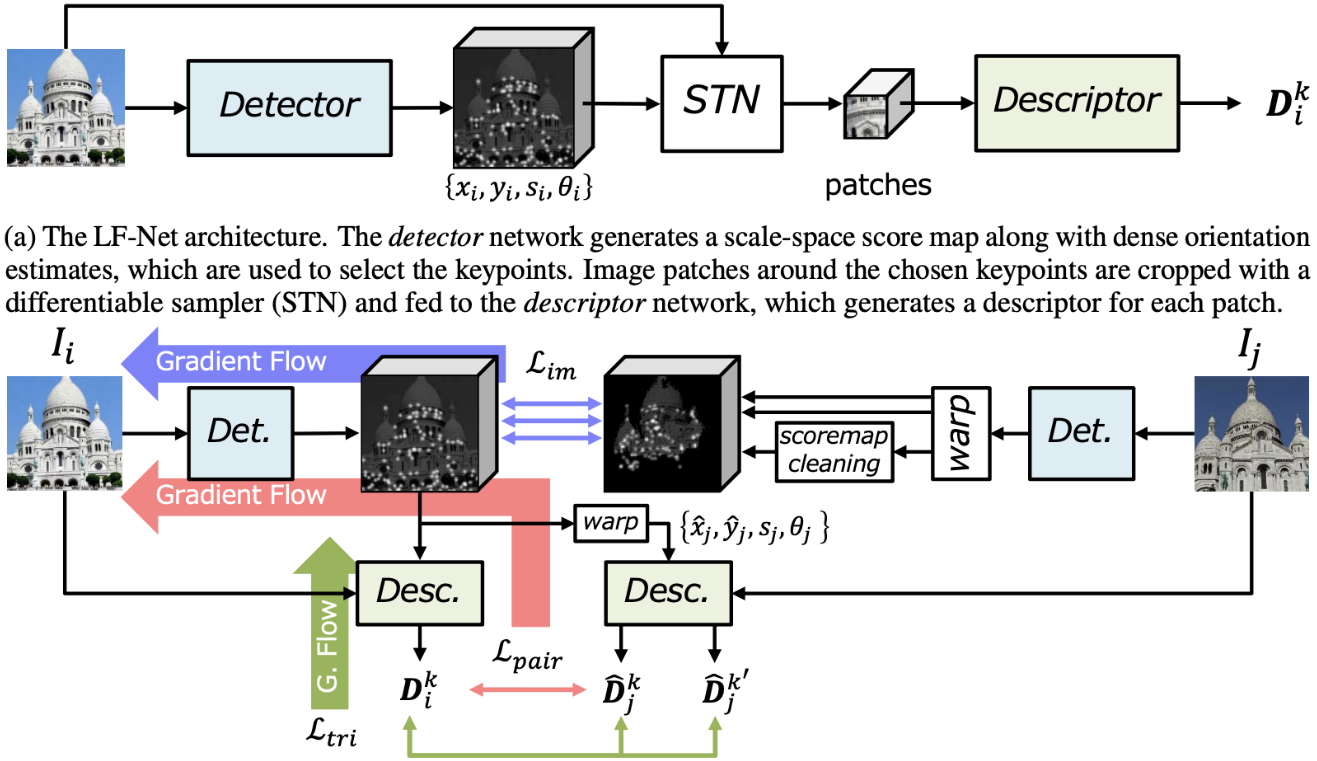
$$\mathcal{L}_{\text{detector}}(\mathbf{P}^1, \mathbf{P}^2, \mathbf{P}^3, \mathbf{P}^4) = \gamma \mathcal{L}_{class}(\mathbf{P}^1, \mathbf{P}^2, \mathbf{P}^3, \mathbf{P}^4) + \mathcal{L}_{pair}(\mathbf{P}^1, \mathbf{P}^2)$$

$$\mathcal{L}_{\text{class}}(\mathbf{P}^1, \mathbf{P}^2, \mathbf{P}^3, \mathbf{P}^4) = \sum_{i=1}^{4} \alpha_i \max\left(0, \left(1 - \text{softmax}\left(f_\mu\left(\mathbf{P}^i\right)\right) y_i\right)\right)^2$$

$$\begin{aligned} \mathcal{L}_{\text{pair}}(\mathbf{P}^1, \mathbf{P}^2) = \| \; & h_\rho(G(\mathbf{P}^1, \text{softargmax}(f_\mu(\mathbf{P}^1)))) - \\ & h_\rho(G(\mathbf{P}^2, \text{softargmax}(f_\mu(\mathbf{P}^2)))) \quad \|_2 \,. \end{aligned}$$

LF-Net: Learning Local Features from Images

LF-Net



(a) The LF-Net architecture. The *detector* network generates a scale-space score map along with dense orientation estimates, which are used to select the keypoints. Image patches around the chosen keypoints are cropped with a differentiable sampler (STN) and fed to the *descriptor* network, which generates a descriptor for each patch.

(b) For training we use a *two-branch* LF-Net, containing two identical copies of the network, processing two corresponding images I_i and I_j. Branch j (right) is used to generate a supervision signal for branch i (left), created by warping the results from i to j. As this is not differentiable, we optimize only over branch i, and update the network copy for branch j in the next iteration. We omit the samplers in this figure, for simplicity.

Figure 1: (a) The Local Feature Network (LF-Net). (b) Training with two LF-Nets.

Detector

- Output of the detectors:
  Score map S + scale map s + orientation map θ

- Loss function 1:

$$\mathcal{L}_{im}(\mathbf{S}_i, \mathbf{S}_j) = |\mathbf{S}_i - g(w(\mathbf{S}_j))|^2 \quad .$$

W(.): warp (变形的参数由sfm模型给出真值)
g(.): clean score map
要注意的是: j branch是不可导的, 借鉴Q-learning的思想用来给出上轮训练的预测结果.
目的: 强调detection repeatablity

Detector

- Loss function2
  从branch i detection score map中选择top K个点. 利用SFM模型找到它们在j图中的对应点.

$$\mathcal{L}_{pair}(\mathbf{D}_i^k, \hat{\mathbf{D}}_j^k) = \sum_k |\mathbf{D}_i^k - \hat{\mathbf{D}}_j^k|^2 \ .$$

- 一个问题:
  为什么这个描述子差异的项会放在这作为detection loss呢?
  ---- 因为STN 参考前面那篇LIFT, 也是假定descriptor网路足够好的条件下, 去训练detector

- Loss function 3:

$$\mathcal{L}_{geom}(s_i^k, \theta_i^k, \hat{s}_i^k, \hat{\theta}_j^k) = \lambda_{ori} \sum_k |\theta_i^k - \hat{\theta}_j^k|^2 + \lambda_{scale} \sum_k |s_i^k - \hat{s}_j^k|^2 \ ,$$

Descriptor

$$\mathcal{L}_{tri}(\mathbf{D}_i^k, \hat{\mathbf{D}}_j^k, \hat{\mathbf{D}}_j^{k'}) = \sum_k \max\left(0, |\mathbf{D}_i^k - \hat{\mathbf{D}}_j^k|^2 - |\mathbf{D}_i^k - \hat{\mathbf{D}}_j^{k'}|^2 + C\right) .$$

对比LIFT

- Two-branch architecture
  而且特点是: 只有一个branch是真正传导梯度的, 另外一个branch仅仅用来提供上一次迭代的结果.
- Lfnet detection 输出中含有scale, 且直接输出orientation信息, 而LIFT则使用一个单独的网络预测orientation.
- LIFT的训练流程较为复杂, descriptor -》 orientation -〉detection
  Lfnet 则是一种end-to-end的训练模式
- LIFT detector的训练特别依赖descriptor的训练好坏
  但是Lfnet由于使用SFM的真值, 引入了pairwise loss, 所以理论上detection结果更好.

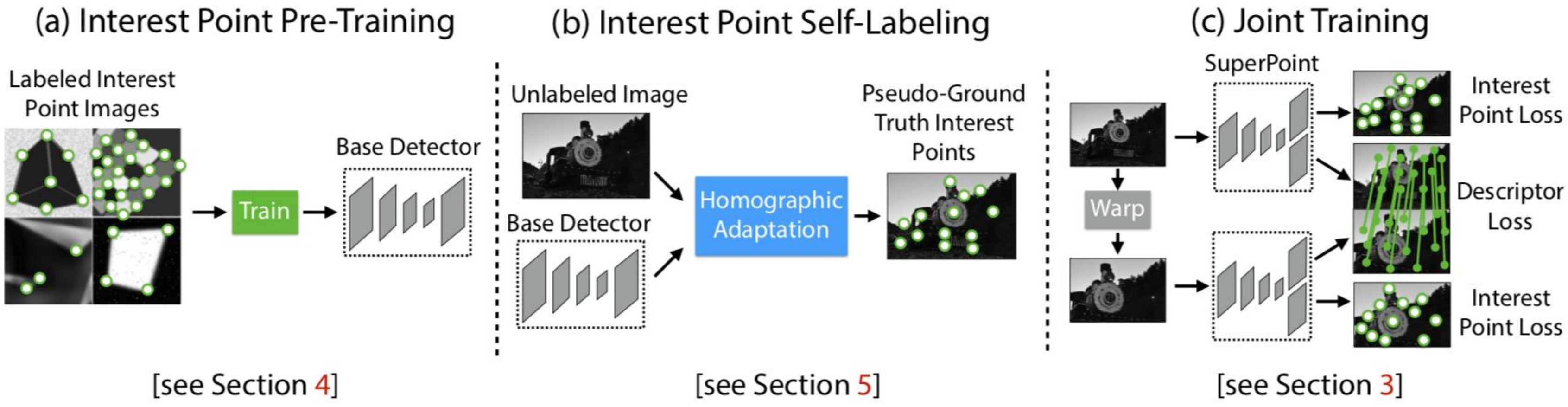
Superpoint
Self-Supervised Interest Point Detection and Description

Motivation

- Problem with strong supervision

The notion of interest point detection is ***semantically ill-defined***. Thus training convolution neural networks with strong supervision of interest points is non-trivial.[关键点定义的模糊性]

Overview



1. 在人工合成的数据集上训练一个简单的检测器 [有监督]
2. Domain adaption: 从人工数据集转移到MS-COCO数据集, 得到该数据集上的伪标签;
3. 利用伪标签训练Detector and descriptor

Synthetic pre-training

- Synthetic shapes

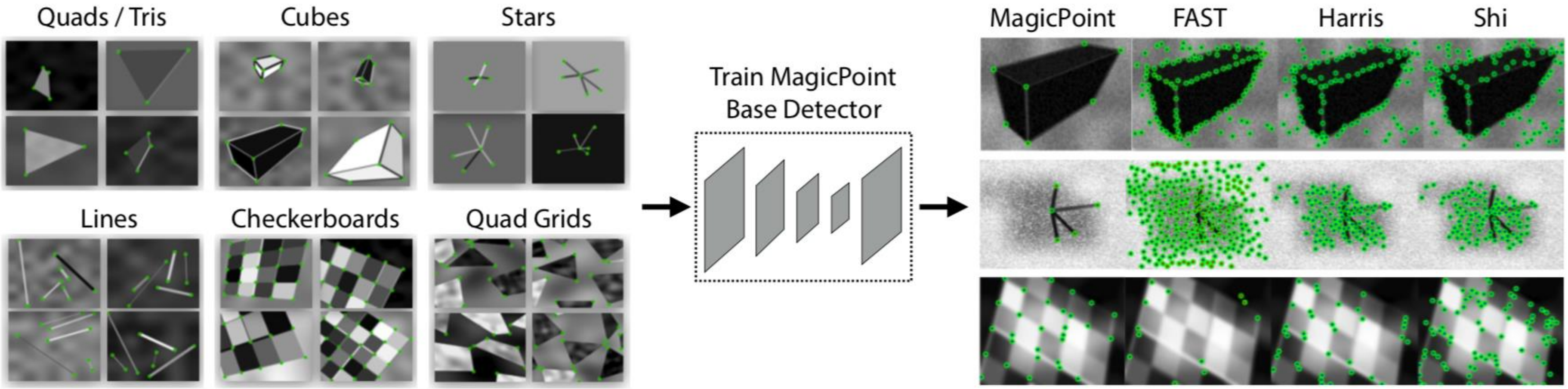


Figure 4. **Synthetic Pre-Training.** We use our Synthetic Shapes dataset consisting of rendered triangles, quadrilaterals, lines, cubes, checkerboards, and stars each with ground truth corner locations. The dataset is used to train the MagicPoint convolutional neural network, which is more robust to noise when compared to classical detectors.

Benefit: 这样的数据集中, key-points的定义是没有任何歧义的

Homographic Adaptions

- Homography

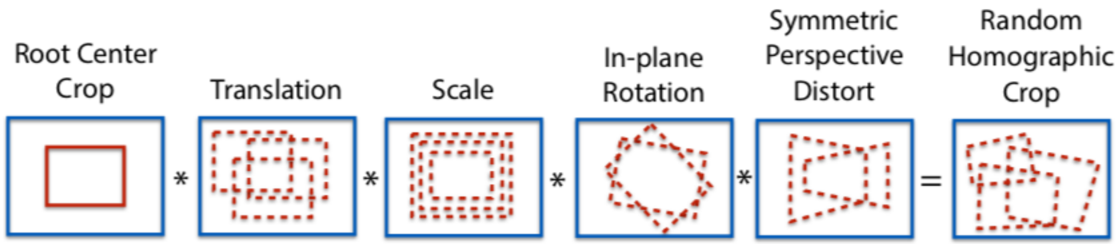


Figure 6. **Random Homography Generation.** We generate random homographies as the composition of less expressive, simple transformations.

仍然是一种线性变换可以用3x3矩阵表示:
2D平面内的仿射变换+3D viewpoint改变

Homographic Adaptions

- Domain Adaption formulation

$$\mathbf{x} = f_\theta(I).$$

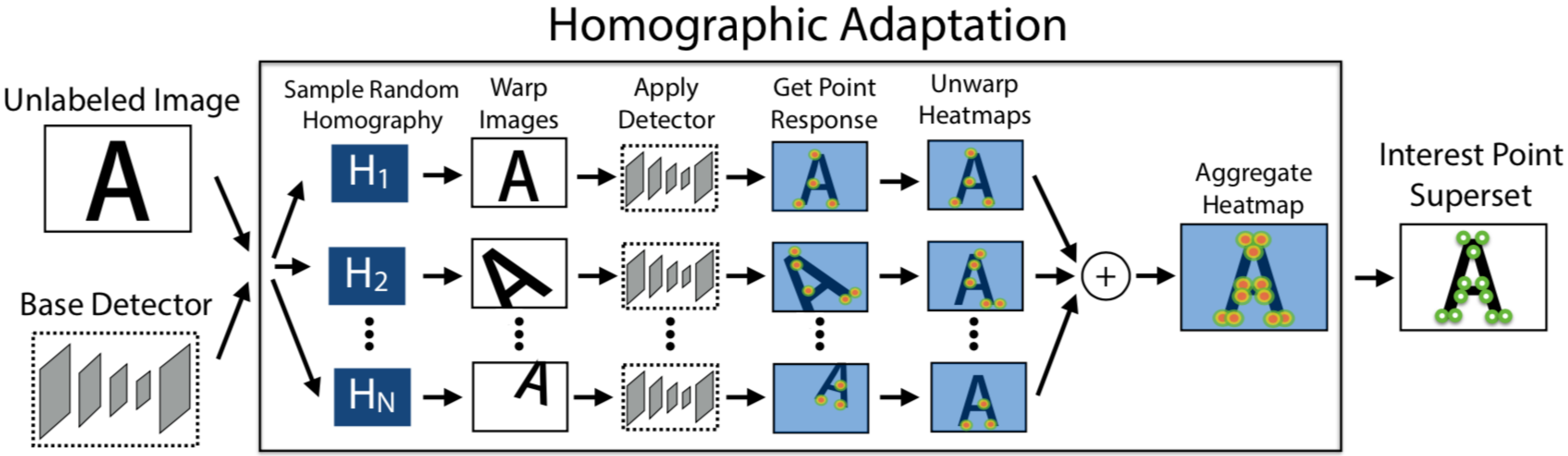
$$\mathcal{H}\mathbf{x} = f_\theta(\mathcal{H}(I)),$$

$$\mathbf{x} = \mathcal{H}^{-1} f_\theta(\mathcal{H}(I)).$$

$$\hat{F}(I; f_\theta) = \frac{1}{N_h} \sum_{i=1}^{N_h} \mathcal{H}_i^{-1} f_\theta(\mathcal{H}_i(I)).$$

Homographic Adaptions

- Domain Adaption diagram



Why?
第一步通过人工数据集训练得到的检测器倾向于检测到更少的关键点,
通过Homographic adaption这种方式可以得到更多的关键点

Homographic Adaptions

- Iterative homographic adaption

上述过程可以迭代:

一次homographic adaption可以将输入的detector变成一个更好的detecor[得到的关键点更多], 多次迭代得到最终的detector.

猜测: 直至检测得到的关键点数目饱和的时候终止

截止至目前: 只有detector

Superpoint

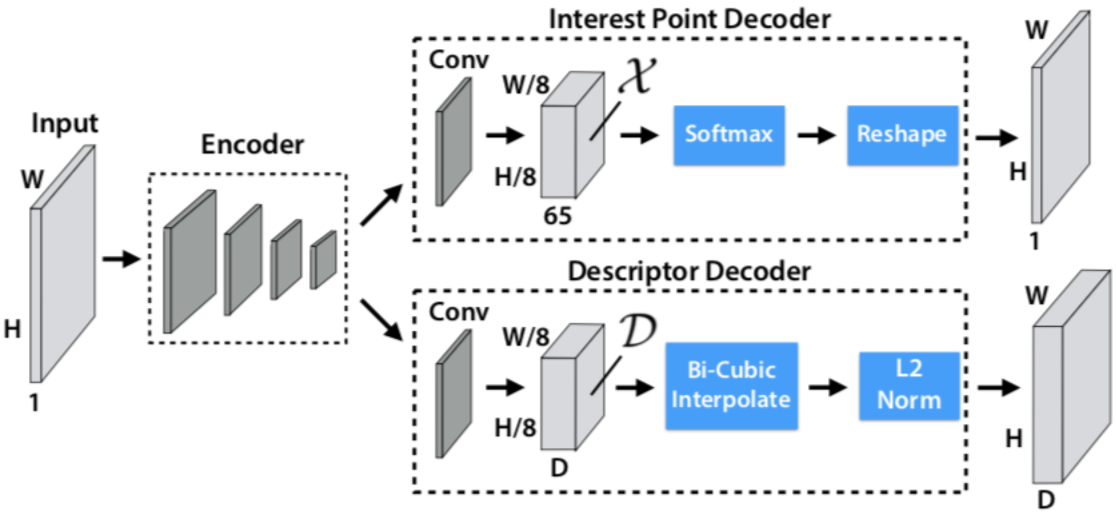


Figure 3. **SuperPoint Decoders**. Both decoders operate on a shared and spatially reduced representation of the input. To keep the model fast and easy to train, both decoders use non-learned upsampling to bring the representation back to $\mathbb{R}^{H \times W}$.

SuperPoint

- Loss function

$$\mathcal{L}(\mathcal{X},\mathcal{X}',\mathcal{D},\mathcal{D}';Y,Y',S) = \\ \mathcal{L}_p(\mathcal{X},Y) + \mathcal{L}_p(\mathcal{X}',Y') + \lambda\mathcal{L}_d(\mathcal{D},\mathcal{D}',S).$$

$$\mathcal{L}_p(\mathcal{X},Y) = \frac{1}{H_cW_c}\sum_{\substack{h=1\\w=1}}^{H_c,W_c} l_p(\mathbf{x}_{hw};y_{hw}),$$

$$\mathcal{L}_d(\mathcal{D},\mathcal{D}',S) = \\ \frac{1}{(H_cW_c)^2}\sum_{\substack{h=1\\w=1}}^{H_c,W_c}\sum_{\substack{h'=1\\w'=1}}^{H_c,W_c} l_d(\mathbf{d}_{hw},\mathbf{d}'_{h'w'};s_{hwh'w'})$$

$$s_{hwh'w'} = \begin{cases} 1, & \text{if } ||\widehat{\mathcal{H}\mathbf{p}_{hw}} - \mathbf{p}_{h'w'}|| \leq 8 \\ 0, & \text{otherwise} \end{cases}$$

$$l_d(\mathbf{d},\mathbf{d}';s) = \lambda_d * s * \max(0, m_p - \mathbf{d}^T\mathbf{d}') \\ +(1-s) * \max(0, \mathbf{d}^T\mathbf{d}' - m_n).$$

DELF: Large-Scale Image Retrieval with Attentive Deep Local Features

Background

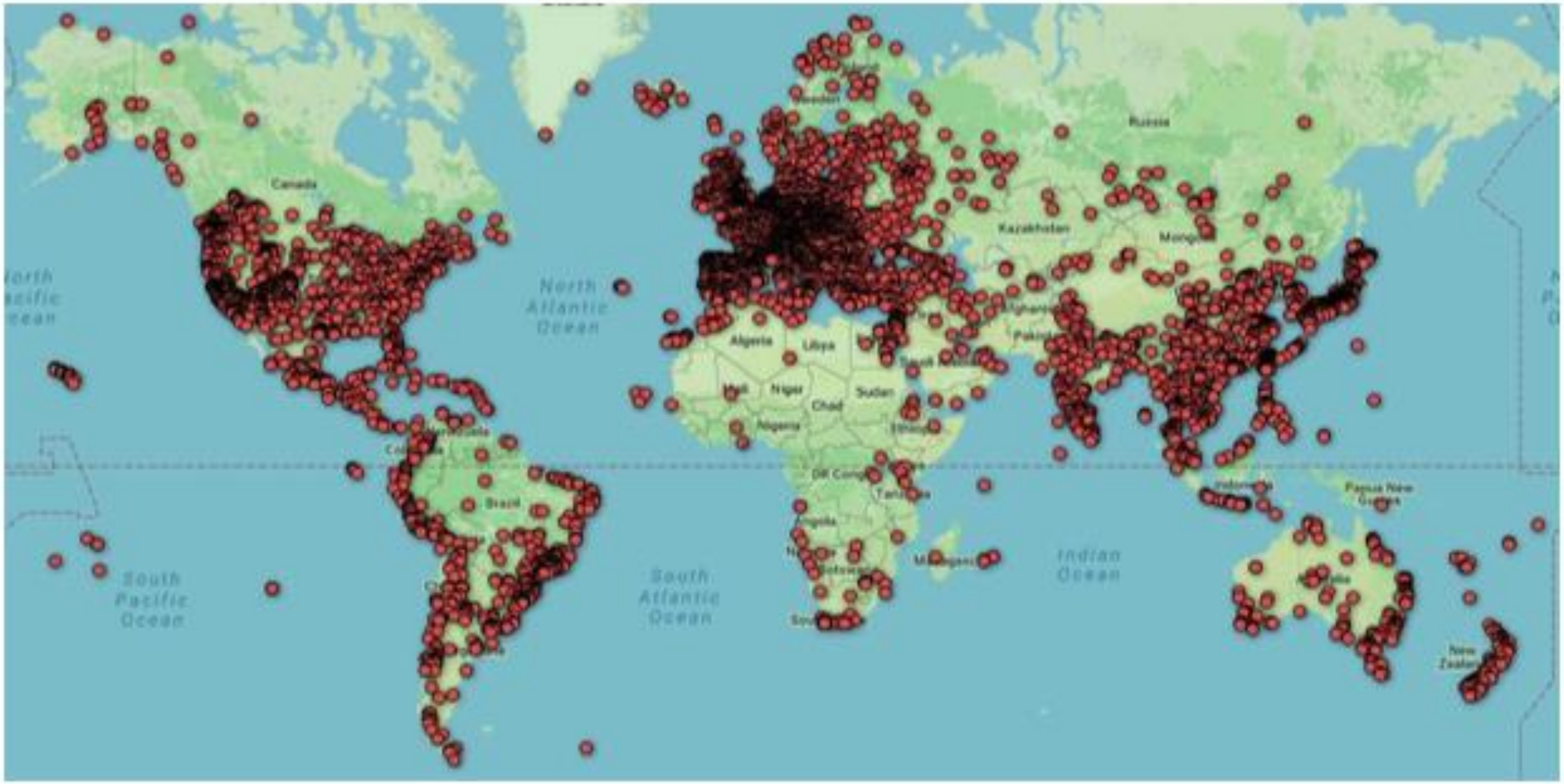


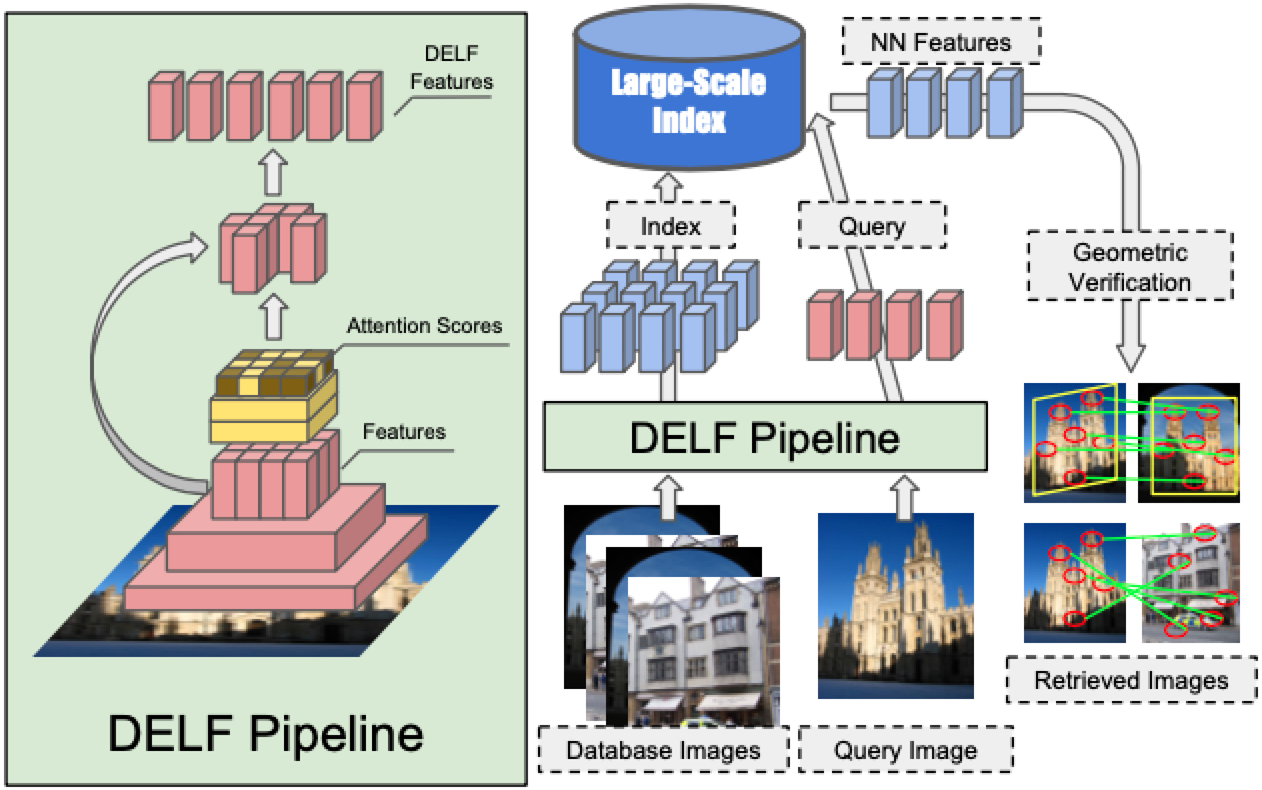
Figure 3: Image geolocation distribution of our Google-Landmarks dataset. The landmarks are located in 4,872 cities in 187 countries.

Landmark set

(a) Sample database images

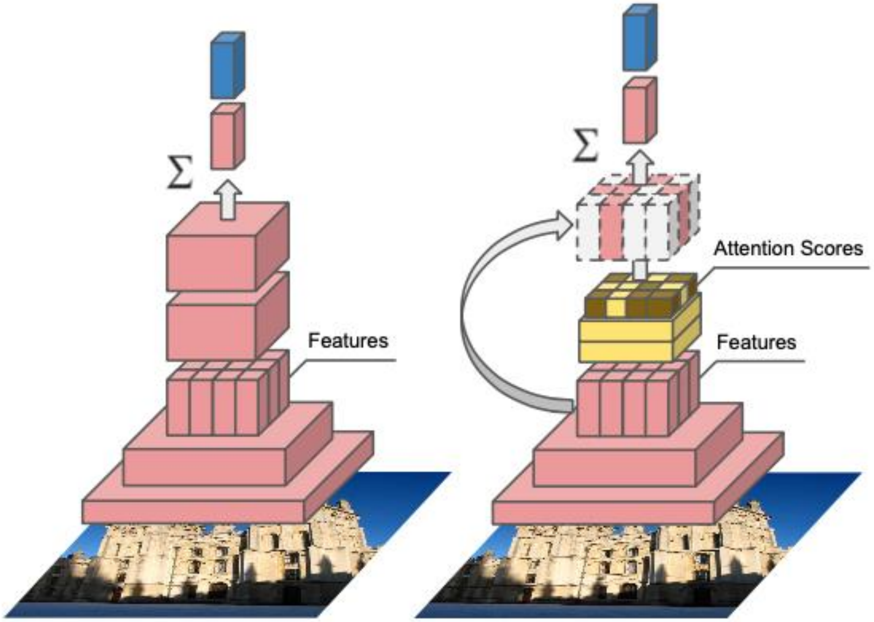
Summary

Solve image retrieval using local features instead of global descriptors.



Training procedure

1. Use pre-trained ResNet
2. Fine tune it on landmark classification problem
3. Train attention module



(a) Descriptor Fine-tuning (b) Attention-based Training

Retrieval process

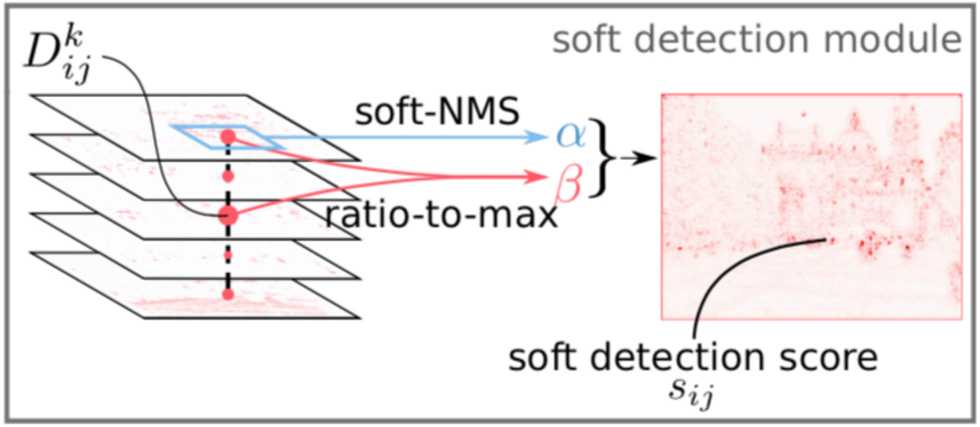
- Representation of a query image: a set of descriptors.
- for each local descriptor, perform KNN search.
- Use the inliers of RANSAC as the matching score.

D2-Net
Detection-and-Description of local features

贺珂

D2net

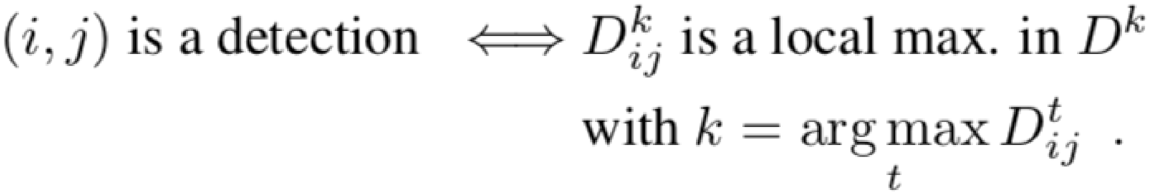
有意思的点:
之前的pipeline都是一种detection then description的流程.
但是本文是detection and description, 甚至某种程度上是 description then detection.



Define keypoint from descriptors

由于这个领域“特征点”没有明确的定义, 所以本文给出了自己的定义:

当它的channel-wise最大响应同时也是空间维度的局部最大响应时, 它是一个特征点.



$$(i, j) \text{ is a detection} \iff D^k_{ij} \text{ is a local max. in } D^k$$
$$\text{with } k = \arg\max_t D^t_{ij} \quad .$$

Define keypoint from descriptors

- Soft feature detection [***train phase***]
  - Soft NMS and ratio-to-max function

$$\alpha_{ij}^{k} = \frac{\exp\left(D_{ij}^{k}\right)}{\sum_{(i',j')\in\mathcal{N}(i,j)} \exp\left(D_{i'j'}^{k}\right)}$$

$$\beta_{ij}^{k} = D_{ij}^{k} / \max_{t} D_{ij}^{t} \quad .$$

$$\gamma_{ij} = \max_{k}\left(\alpha_{ij}^{k}\beta_{ij}^{k}\right)$$

$$s_{ij} = \gamma_{ij} \Big/ \sum_{(i',j')} \gamma_{i'j'}$$