NinjaDesc: Content-Concealing Visual Descriptors via Adversarial Learning
Supplementary Material

Tony Ng[1,2] Hyo Jin Kim[1*] Vincent T. Lee[1] Daniel DeTone[1] Tsun-Yi Yang[1]
Tianwei Shen[1] Eddy Ilg[1] Vassileios Balntas[1] Krystian Mikolajczyk[2] Chris Sweeney[1]
[1]Reality Labs, Meta [2]Imperial College London

We first provide a comparison of our NinjaDesc and the base descriptor on the 3D reconstruction task using SfM (Sec. A). Next, we report the full HPatches results using HardNet [7] and SIFT [6] as the base descriptors (Sec. B). In addition to our results on Aachen-Day-Night v1.1 in the main paper, we also provide our results on Aachen-Day-Night v1.0 (Sec. C). Finally, we illustrate the detailed architecture for the inverse models (Sec. E).

A. 3D Reconstruction

Table 1 shows a quantitative comparison of our content-concealing NinjaDesc and the base descriptor SOSNet [12] on the SfM reconstruction task using the landmarks dataset for local feature benchmarking [11]. As can be seen, decrease in the performance for our content-concealing NinjaDesc is only marginal for all metrics.

B. Full HPatches results for HardNet and SIFT

Figure 1 illustrates our full evaluation results on HPatches using HardNet [7] and SIFT [6] as the base descriptors for NinjaDesc, in addition to the results using SOSNet [12] provided in the main paper. Similar to the results for SOSNet [12], we observe little drop in accuracy for NinjaDesc overall compared to the original base descriptors, ranging from low ($\lambda = 0.1$) to high ($\lambda = 2.5$) privacy parameters.

C. Evaluation on Aachen-Day-Night v1.0

In Table 2 of the main paper, we report the result of NinjaDesc on Aachen-Day-Night v1.1 dataset. The v1.1 is updated with more accurate ground-truths compared to the older v1.0. Because Dusmanu *et al.* [3] performed evaluation on the v1.0, we also provide our results on v1.0 in Table 2 for better comparison.

*Corresponding author.

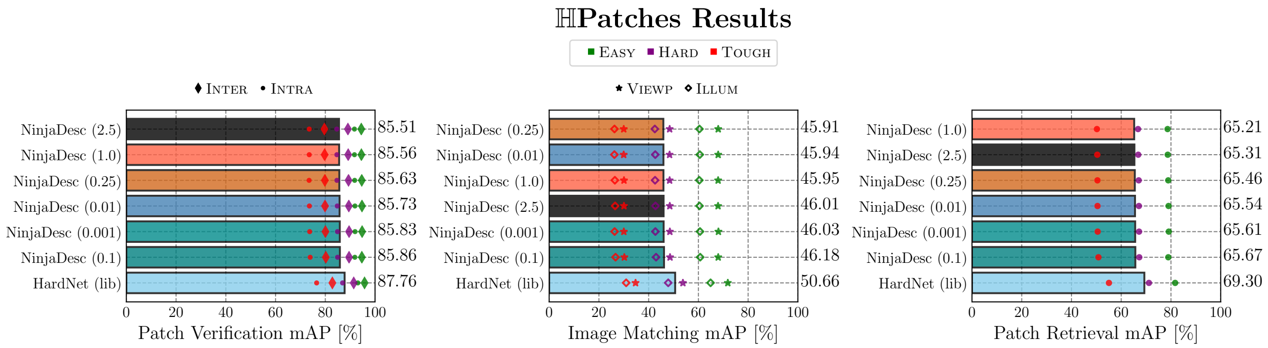
Dataset	Method	Reg. images	Sparse points	Observations	Track length	Reproj. error
South-Building 128 images	SOSNet	128	101,568	638,731	6.29	0.56
	NinjaDesc (1.0)	128	105,780	652,869	6.17	0.56
	NinjaDesc (2.5)	128	105,961	653,449	6.17	0.56
Madrid Metropolis 1344 images	SOSNet	572	95,733	672,836	7.03	0.62
	NinjaDesc (1.0)	566	94,374	668,148	7.08	0.64
	NinjaDesc (2.5)	564	94,104	667,387	7.09	0.63
Gendarmenmarkt 1463 images	SOSNet	1076	246,503	1,660,694	6.74	0.74
	NinjaDesc (1.0)	1087	312,469	1,901,060	6.08	0.75
	NinjaDesc (2.5)	1030	340,144	1,871,726	5.50	0.77
Tower of London 1463 images	SOSNet	825	200,447	1,733,994	8.65	0.62
	NinjaDesc (1.0)	797	198,767	1,727,785	8.69	0.62
	NinjaDesc (2.5)	837	218,888	1,792,908	8.19	0.64

Table 1. 3D reconstruction statistics on the local feature evaluation benchmark [11]. Number in parenthesis is the privacy parameter λ.

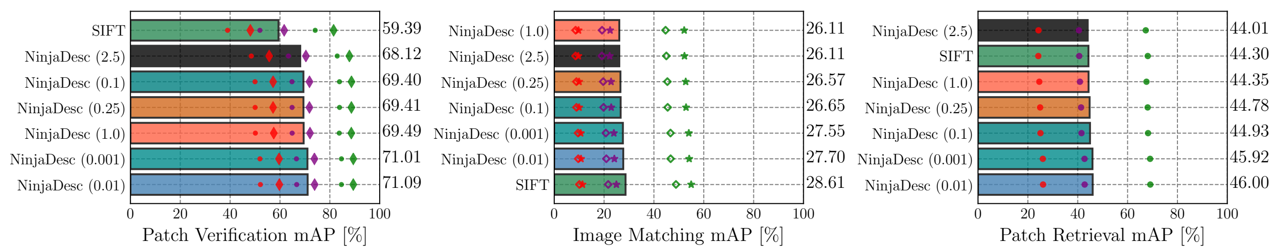
D. Additional content-concealment experiments

1. Nearest-neighbour attack. Two examples of nearest-neighbour (NN) attack similar to that in [3] using a database of 128,000 existing descriptors are shown in Fig. 2. In both NN attack scenarios, the reconstruction is significantly deteriorated, as it is non-trivial to compute distances between the two spaces, *cf*. oracle attack analysis below. Note we use $\lambda = 2.5$ for all our experiments.

2. Oracle attack distance analysis. The distances to the original descriptor using the oracle attack following [3] is plotted in black in Fig. 3. We also show an alternative oracle (red dotted), which differs from [3] in that the K neighbours are first matched using the NinjaDesc database,



(a) HardNet Base Descriptor



(b) SIFT Base Descriptor

Figure 1. HPatches evaluation results. For each base descriptor (HardNet [7] and SIFT [6]), we compare with NinjaDesc, with 5 different levels of privacy parameter λ (indicated by the number in parenthesis). All results are from models trained on the *liberty* subset of the UBC patches [4] dataset, apart from SIFT which is handcrafted, and we use the Kornia [9] GPU implementation evaluated on 32×32 patches.

Query	NNs	Method	Accuracy @ Thresholds (%) 0.25m, 2°	0.5m, 5°	5.0m, 10°
		Base Desc	SOS / Hard / SIFT	SOS / Hard / SIFT	SOS / Hard / SIFT
Day (824)	20	Raw	85.1 / 85.4 / 84.3	92.7 / 93.1 / 92.7	97.3 / 98.2 / 97.6
		$\lambda = 0.1$	85.4 / 84.7 / 82.0	92.5 / 91.9 / 91.1	97.5 / 96.8 / 96.4
		$\lambda = 1.0$	84.7 / 84.3 / 82.9	92.4 / 91.9 / 91.0	97.2 / 96.7 / 96.1
		$\lambda = 2.5$	84.6 / 83.7 / 82.5	92.4 / 92.0 / 91.0	97.1 / 96.8 / 96.0
	50	Raw	85.9 / 86.8 / 86.0	92.5 / 93.7 / 94.1	97.3 / 98.1 / 98.2
		$\lambda = 0.1$	85.2 / 85.2 / 84.2	92.2 / 92.4 / 91.4	97.1 / 97.1 / 96.6
		$\lambda = 1.0$	84.7 / 85.7 / 83.4	92.2 / 92.6 / 91.6	97.2 / 96.7 / 96.7
		$\lambda = 2.5$	85.6 / 85.3 / 83.6	92.7 / 91.7 / 91.1	97.3 / 96.8 / 96.2
Night (98)	20	Raw	51.0 / 57.2 / 55.1	65.3 / 68.4 / 67.3	70.4 / 76.5 / 74.5
		$\lambda = 0.1$	51.0 / 45.9 / 45.9	62.2 / 56.1 / 54.1	68.4 / 62.2 / 63.3
		$\lambda = 1.0$	50.0 / 43.9 / 44.9	62.2 / 54.1 / 56.1	66.3 / 62.2 / 64.3
		$\lambda = 2.5$	48.0 / 44.9 / 44.9	58.2 / 59.2 / 52.0	65.3 / 65.3 / 62.2
	50	Raw	48.0 / 51.0 / 54.1	59.2 / 64.3 / 65.3	65.3 / 68.4 / 74.5
		$\lambda = 0.1$	41.8 / 39.8 / 41.8	52.0 / 51.0 / 52.0	60.2 / 56.1 / 60.2
		$\lambda = 1.0$	43.9 / 39.8 / 43.9	54.1 / 50.0 / 54.1	63.3 / 58.2 / 63.3
		$\lambda = 2.5$	42.9 / 40.8 / 42.9	52.0 / 50.0 / 52.0	61.2 / 56.1 / 58.2

Table 2. Visual localization results on Aachen-Day-Night v1.0 [10]. 'Raw' corresponds to the base descriptor in each column, followed by three λ vales (0.1, 1.0, 2.5) for NinjaDesc.

then their corresponding SOSNet descriptor pairings are retrieved. For completeness, we also plot the results of only using NinjaDesc descriptors as the database (blue dashed).

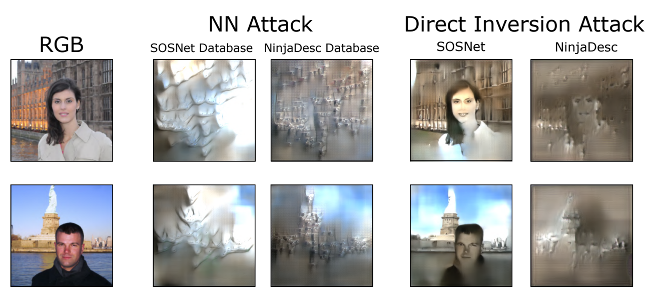


Figure 2. Examples of NN attack. For NN attack, we show results using SOSNet and our NinjaDesc descriptors to form the database.

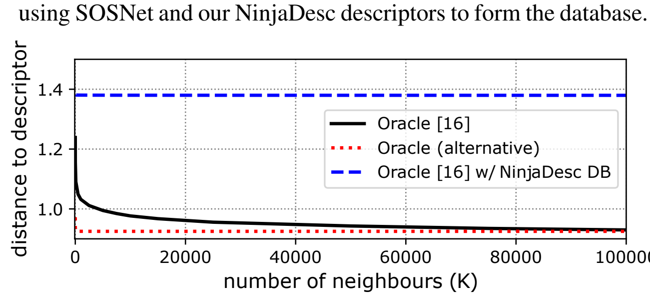


Figure 3. Distances to the original descriptor (SOSNet) of the nearest-neighbour retrieved by three variants of the oracle attack.

We observe that the distance decreases as K increases for SOSNet database like Fig. 6 in [3]. However, we argue that this alone does not validate manifold folding. Rather, as K increases we approach the limit of the distance to the real NN of the original (SOSNet) descriptor, regardless of the private (NinjaDesc) representation. This limit is achieved by the alternative oracle (red dotted), where the closest NinjaDesc (*i.e*. the corresponding SOSNet) database descriptor is always retrieved, for most K values. If the oracle in [3] uses the NinjaDesc database (blue dashed), the distance remains large. This is because unlike [3], NinjaNet maps the original feature space to a completely new one via learned non-linear transformations, and is thus robust to distance calculation across the two descriptor spaces.

Fig. 4 shows how our reconstruction improves as K increases in oracle attack [3]. Still, even with very large K, it is visibly worse than that from direct inversion or the original image. For the oracle with NinjaDesc database (last column), the reconstruction is highly privacy-preserving.

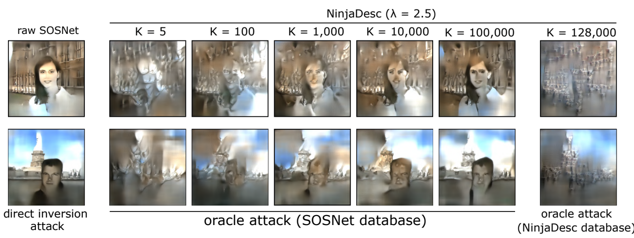


Figure 4. Examples of oracle attack w.r.t. num. of neighbours K.

As noted in [3], an oracle attack is impractical as the attacker does not have access to the original descriptors.

E. Detailed architectures of the descriptor inversion models

UNet. The architecture of the UNet-based descriptor inversion model, which is also used in [1,8], is shown in Figure 5.

UResNet. Figure 6 illustrates the architecture of the descriptor inversion model based on UResNet used for the ablation study in the Section 5.2 of the main paper. The overall "U" shape of UResNet is similar to UNet, but each convolution block is drastically different. We use the 5 stages of ResNet50 [5] (pretrained on ImageNet [2]) {`conv1`, `conv2_x`, `conv3_x`, `conv4_x`, `conv4_x`} as the 5 encoding / down-sampling blocks, except for `conv2_x` we remove the `MaxPool2d` so that each encoding block corresponds to a 1/2 down-sampling in resolution. Since ResNet50 takes in RGB image as input (which has shape of $3 \times h \times w$, whereas the sparse feature maps are of shape $128 \times h \times w$), we pre-process the input with 4 additional basic redisual blocks denoted by `res_conv_block` in Figure 6. The up-sampling decoder blocks (denoted by `up_conv`) are also residual blocks with an addition input up-sampling layer using bilinear interpolation. In contrast to UNet, the skip connections in our UResNet are performed by additions, rather than concatenations.

References

[1] Deeksha Dangwal, Vincent T. Lee, Hyo Jin Kim, Tianwei Shen, Meghan Cowan, Rajvi Shah, Caroline Trippel, Brandon Reagen, Timothy Sherwood, Vasileios Balntas, Armin Alaghi, and Eddy Ilg. Analysis and mitigations of reverse engineering attacks on local feature descriptors. In *BMVC*, 2021.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[3] Mihai Dusmanu, Johannes L Schönberger, Sudipta N Sinha, and Marc Pollefeys. Privacy-preserving visual feature descriptors through adversarial affine subspace embedding. In *CVPR*, 2021.

[4] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz. Multi-view stereo for community photo collections. In *CVPR*, 2007.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[6] David G. Lowe. Distinctive image features from scale-invariant keypoints. In *IJCV*, 2004.

[7] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenović, and Jiřì Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NIPS*, 2017.

[8] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *CVPR*, 2019.

[9] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: an open source differentiable computer vision library for PyTorch. In *WACV*, 2020.

[10] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *CVPR*, 2018.

[11] Johannes L Schonberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *CVPR*, 2017.

[12] Yurun Tian, Xin Yu, Bin Fan, Wu. Fuchao, Huub Heijnen, and Vassileios Balntas. SOSNet: Second order similarity regularization for local descriptor learning. In *CVPR*, 2019.

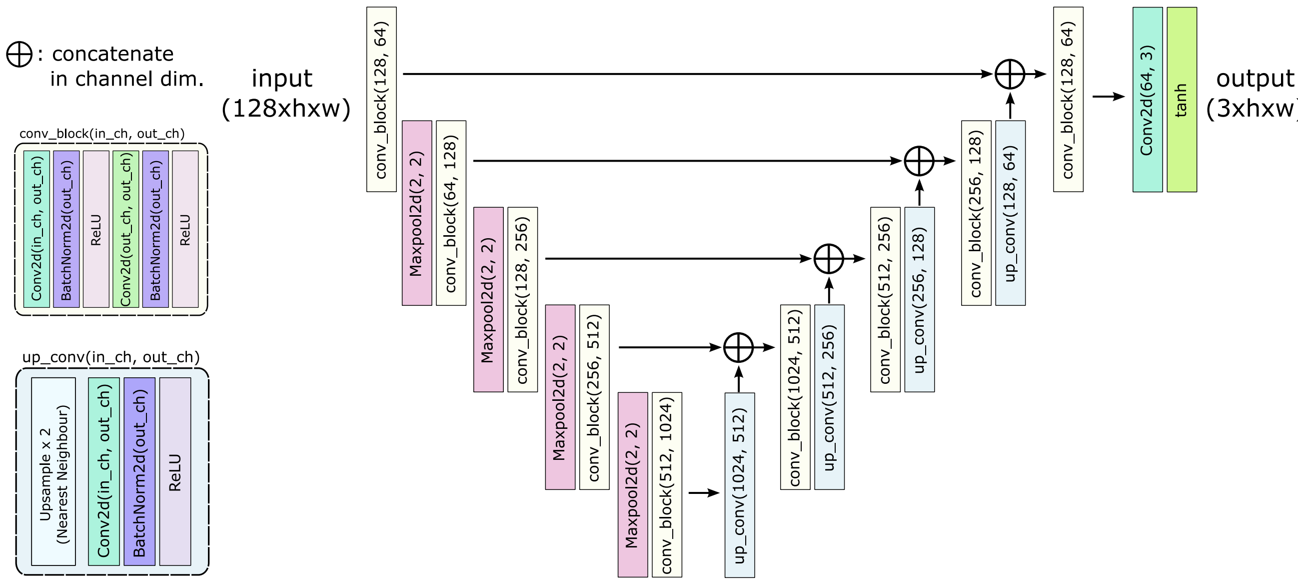


Figure 5. UNet Architecture.

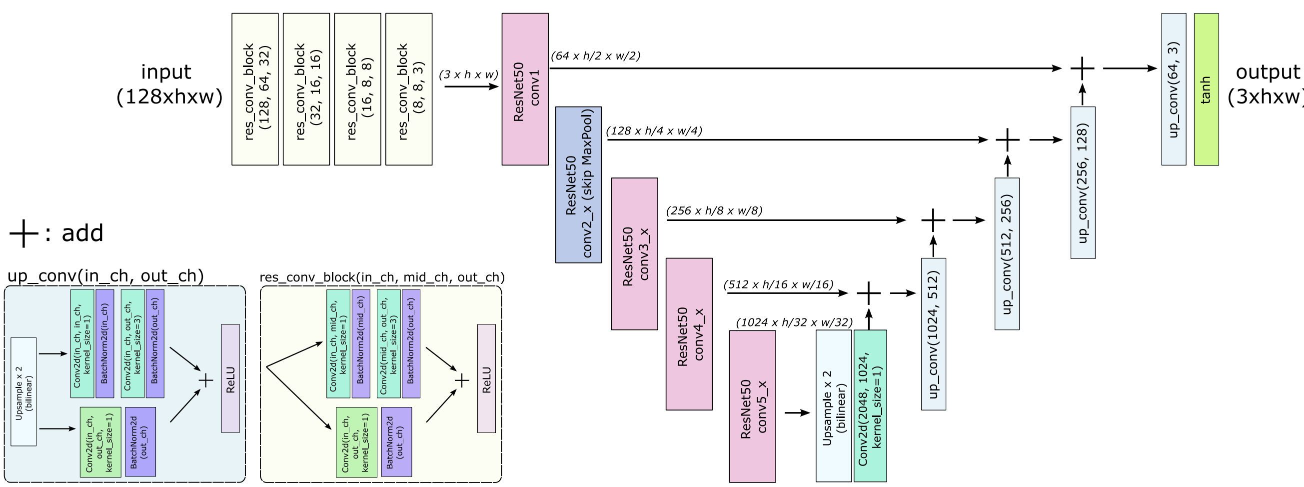


Figure 6. UResNet Architecture.