Generative Domain-Migration Hashing for Sketch-to-Image Retrieval

ECCV 2018 European Conference on Computer Vision
8 – 14 September 2018 | Munich, Germany

Jingyi Zhang †‡, Fumin Shen†, Li Liu‡, Fan Zhu‡, Mengyang Yu⋆, Ling Shao ‡, Heng Tao Shen†, Luc Van Gool⋆
† Center for Future Media and School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu, China
‡ Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
⋆ Computer Vision Lab, ETH Zurich, Switzerland

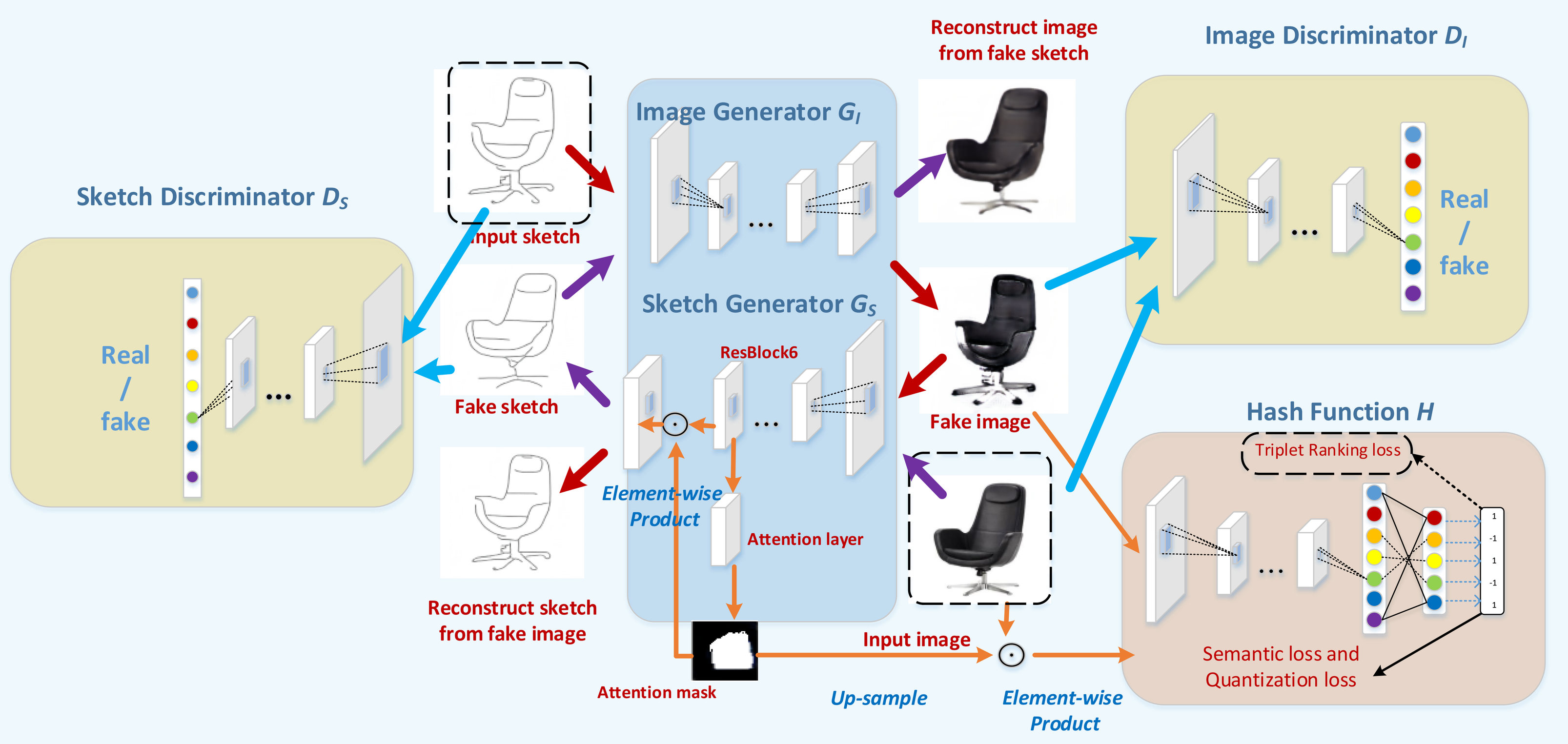
Motivations

Sketch Based Image Retrieval (SBIR) remains a long-standing unsolved problem mainly because of the significant discrepancy between the sketch domain and the image domain. Our goal is to overcome this limitation and propose a new method which can better bridge the domain gap and achieve higher performance.

Contributions

- We for the first time propose a generative model GDH for the hashing-based SBIR problem. Comparing to existing methods, the generative model can essentially improve the generalization capability by migrating sketches into their indistinguishable counterparts in the natural image domain.
- Guided by an adversarial loss and a cycle consistency loss, the optimized binary hashing codes can preserve the semantic consistency across domains. Meanwhile, training GDH does not require the pixel-level alignment across domains, and thus allows generalized and practical applications.
- GDH can improve the category-level SBIR performance over the state-of-the-art hashing-based SBIR method DSH by up to 20.5% on the TU-Berlin Extension dataset, and up to 26.4% on the Sketchy dataset respectively. Meanwhile, GDH can achieve comparable performance with real-valued fine-grained SBIR methods, while significantly reduce the retrieval time and memory cost with binary codes.

Framework



Experiments

In the experiment section, we address the following three questions:

- How does GDH perform as compared to other state-of-the-art binary or real-valued methods for category-level SBIR?
- How does GDH perform as compared to other state-of-the-art real-valued methods for fine-grained SBIR?
- How does each component or constraint contribute to the overall performance of GDH?

Comparison with previous SBIR methods

Methods	Dimension	TU-Berlin Extension MAP	TU-Berlin Extension Retrieval time per query (s)	TU-Berlin Extension Memory cost (MB) (204,489 images)	Sketchy MAP	Sketchy Retrieval time per query (s)	Sketchy Memory cost (MB) (73,002 images)
HOG	1296	0.091	1.43	2.02×10^3	0.115	0.53	7.22×10^2
GF-HOG	3500	0.119	4.13	5.46×10^3	0.157	1.41	1.95×10^3
SHELO	1296	0.123	1.44	2.02×10^3	0.182	0.50	7.22×10^2
LKS	1350	0.157	0.204	2.11×10^3	0.190	0.56	7.52×10^2
Siamese CNN	64	0.322	7.70×10^{-2}	99.8	0.481	2.76×10^{-2}	35.4
SaN	512	0.154	0.53	7.98×10^2	0.208	0.21	2.85×10^2
GN Triplet*	1024	0.187	1.02	1.60×10^3	0.529	0.41	5.70×10^2
3D shape*	64	0.072	7.53×10^{-2}	99.8	0.084	2.64×10^{-2}	35.6
Siamese-AlexNet	4096	0.367	5.35	6.39×10^3	0.518	1.68	2.28×10^3
Triplet-AlexNet	4096	0.448	5.35	6.39×10^3	0.573	1.68 s	2.28×10^3
GDH (Proposed)	32 (bits)	0.563	5.57×10^{-4}	0.78	0.724	2.55×10^{-4}	0.28
	64 (bits)	**0.690**	7.03×10^{-4}	1.56	**0.810**	2.82×10^{-4}	0.56
	128 (bits)	0.659	1.05×10^{-3}	3.12	0.784	3.53×10^{-4}	1.11

"*" denotes that we directly use the public models provided by the original papers without any fine-tuning on the TU-Berlin Extension and Sketchy datasets.

Comparison with cross-modality methods

Method		TU-Berlin Extension 32 bits	TU-Berlin Extension 64 bits	TU-Berlin Extension 128 bits	Sketchy 32 bits	Sketchy 64 bits	Sketchy 128 bits
Cross-Modality Hashing Methods (binary codes)	CMFH	0.149	0.202	0.180	0.320	0.490	0.190
	CMSSH	0.121	0.183	0.175	0.206	0.211	0.211
	SCM-Seq	0.211	0.276	0.332	0.306	0.417	0.671
	SCM-Orth	0.217	0.301	0.263	0.346	0.536	0.616
	CVH	0.214	0.294	0.318	0.325	0.525	0.624
	SePH	0.198	0.270	0.282	0.534	0.607	0.640
	DCMH	0.274	0.382	0.425	0.560	0.622	0.656
	DSH	0.358	0.521	0.570	0.653	0.711	0.783
Cross-View Feature Learning Methods (real-valued vectors)	CCA	0.276	0.366	0.365	0.361	0.555	0.705
	XQDA	0.191	0.197	0.201	0.460	0.557	0.550
	PLSR	0.141 (4096-d)			0.462 (4096-d)		
	CVFL	0.289 (4096-d)			0.675 (4096-d)		
Proposed	**GDH**	**0.563**	**0.690**	**0.651**	**0.724**	**0.811**	**0.784**

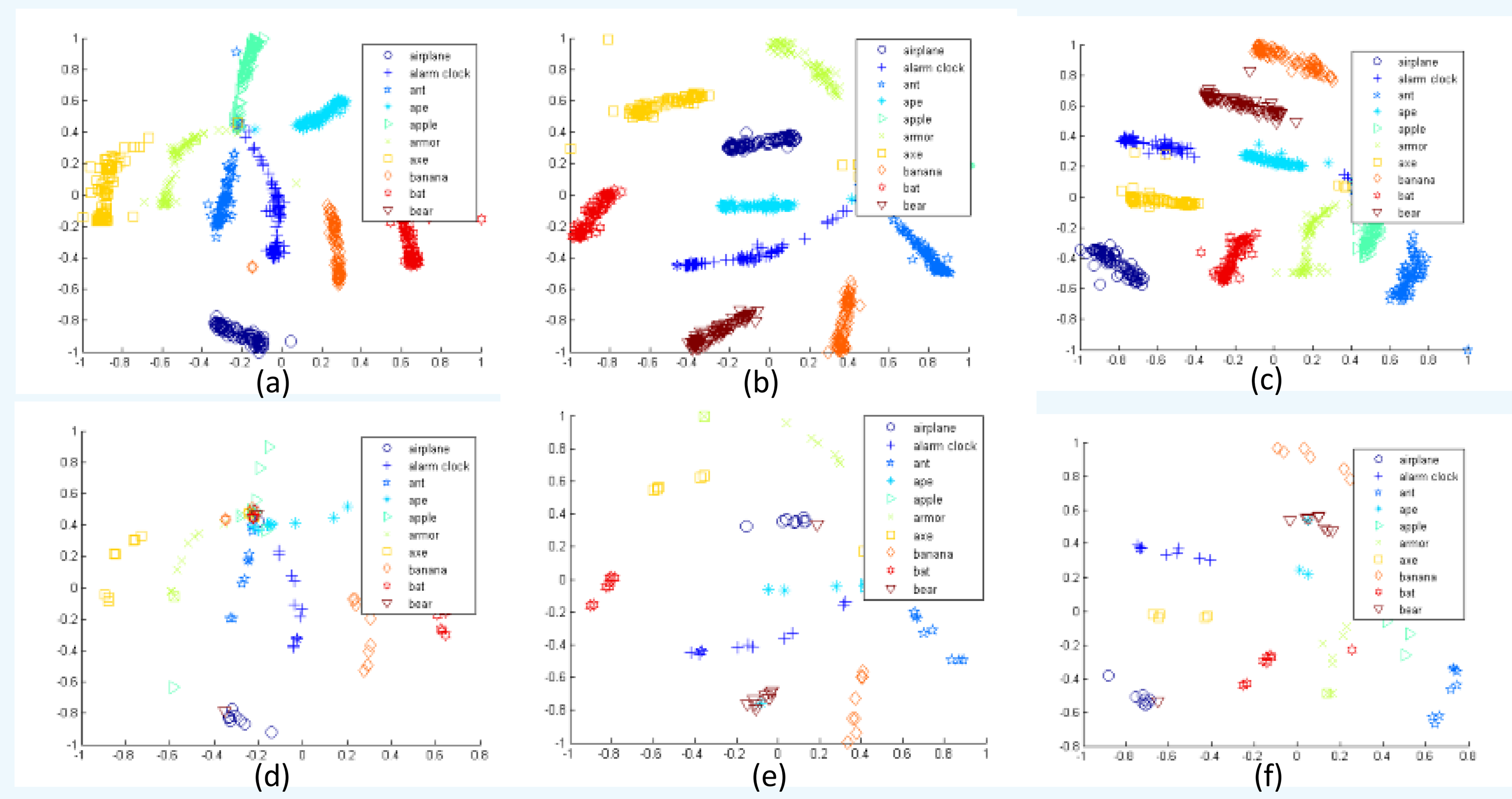
For end-to-end deep methods, raw natural images and sketches are used. For others, 4096-d AlexNet *fc7* image features and 512-d SaN *fc7* sketch features are used. PLSR and CVFL are both based on reconstructing partial data to approximate full data, so the dimensions are fixed to 4096-d.

Comparison with fine-grained method

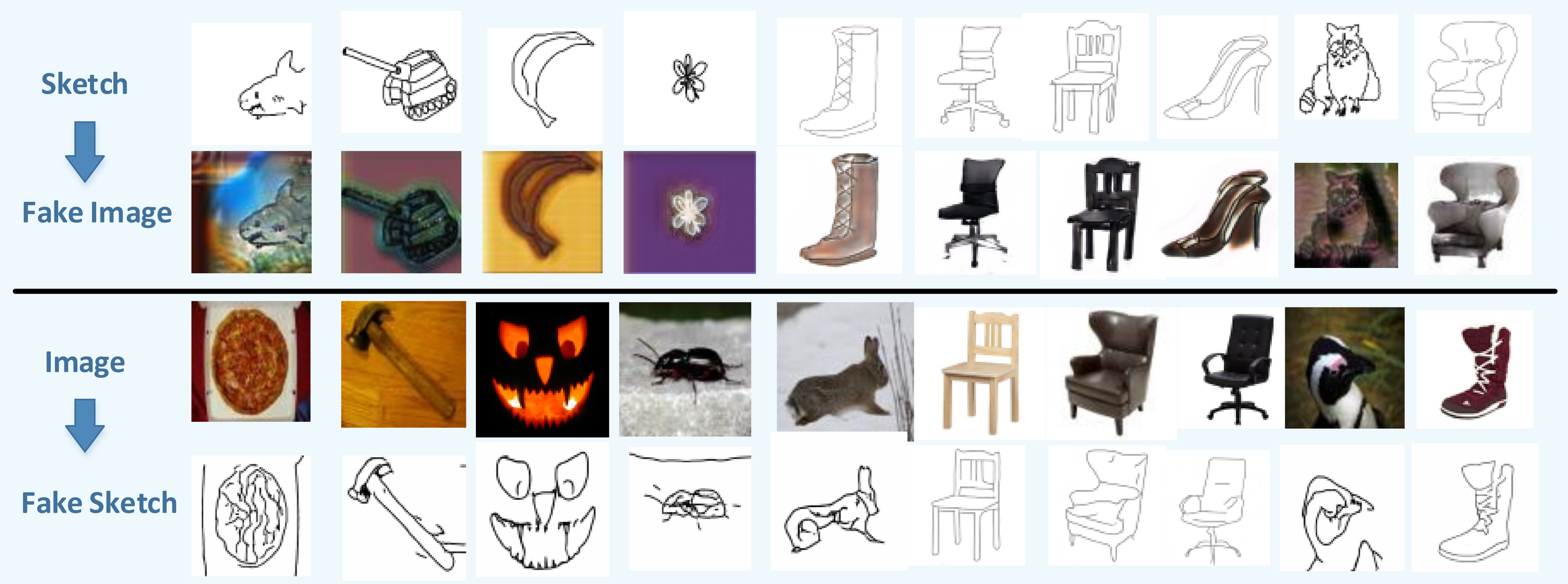
Methods		QMUL-shoes.acc@1	QMUL-shoes.acc@10	QMUL-chairs.@1	QMUL-chairs.@10
Real-valued vectors	BoW-HOG + rankSVM	0.174	0.678	0.289	0.670
	Dense-HOG + rankSVM	0.244	0.652	0.526	0.938
	ISN Deep + rankSVM	0.200	0.626	0.474	0.825
	3DS Deep + rankSVM	0.052	0.217	0.061	0.268
	TSN without data aug.	0.330	0.817	0.644	0.956
	TSN with data aug.	0.391	0.878	0.691	0.979
Binary codes	GDH @ 32-bit	0.286	0.720	0.392	0.876
	GDH @ 64-bit	0.323	0.783	0.556	0.959
	GDH @ 128-bit	0.357	0.843	0.671	0.990

To emphasize the ability of our domain-migration model, data augmentation is not included. Even so, our binary results are competitive and promising.

t-SNE visualization



Visualization of domain-migration networks



Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Project 61502081 and Project 61632007, the Fundamental Research Funds for the Central Universities under Project ZYGX2014Z007.