MIHash: Online Hashing with Mutual Information

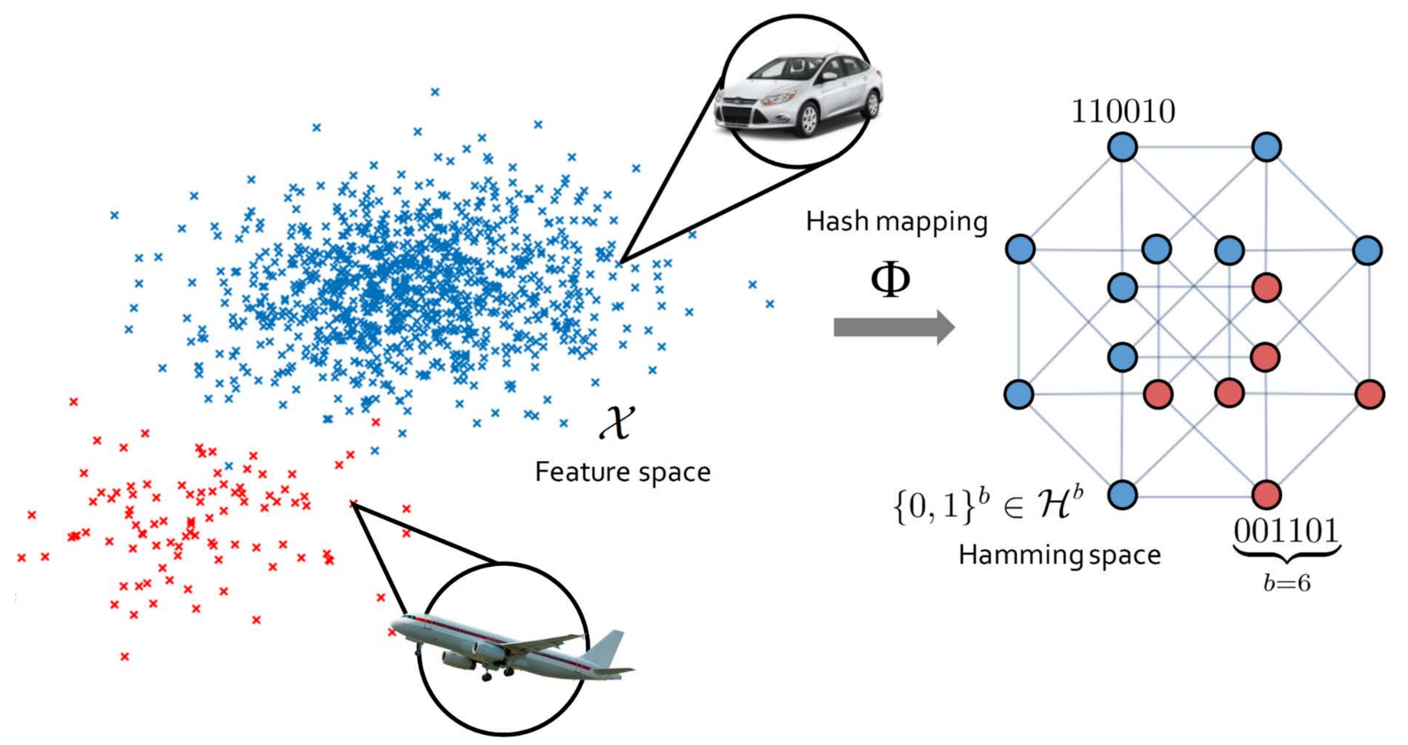
Fatih Cakir*, Kun He*, Sarah Adel Bargal, Stan Sclaroff (*equal contribution)

Computer Science, Boston University {fcakir,hekun,sbargal,sclaroff}@cs.bu.edu

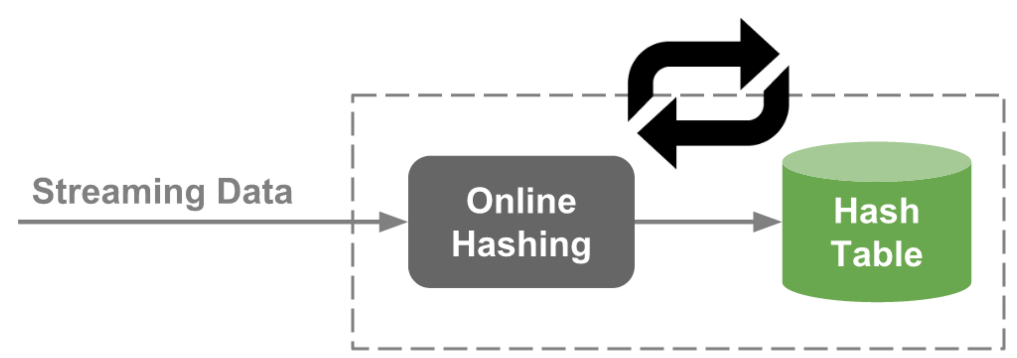
BOSTON UNIVERSITY

ICCV 17 International Conference on Computer Vision 2017

Background



Online Hashing: learn $\Phi : \mathcal{X} \rightarrow \mathcal{H}^b$ with streaming data.



Q: When do we update the hash table?

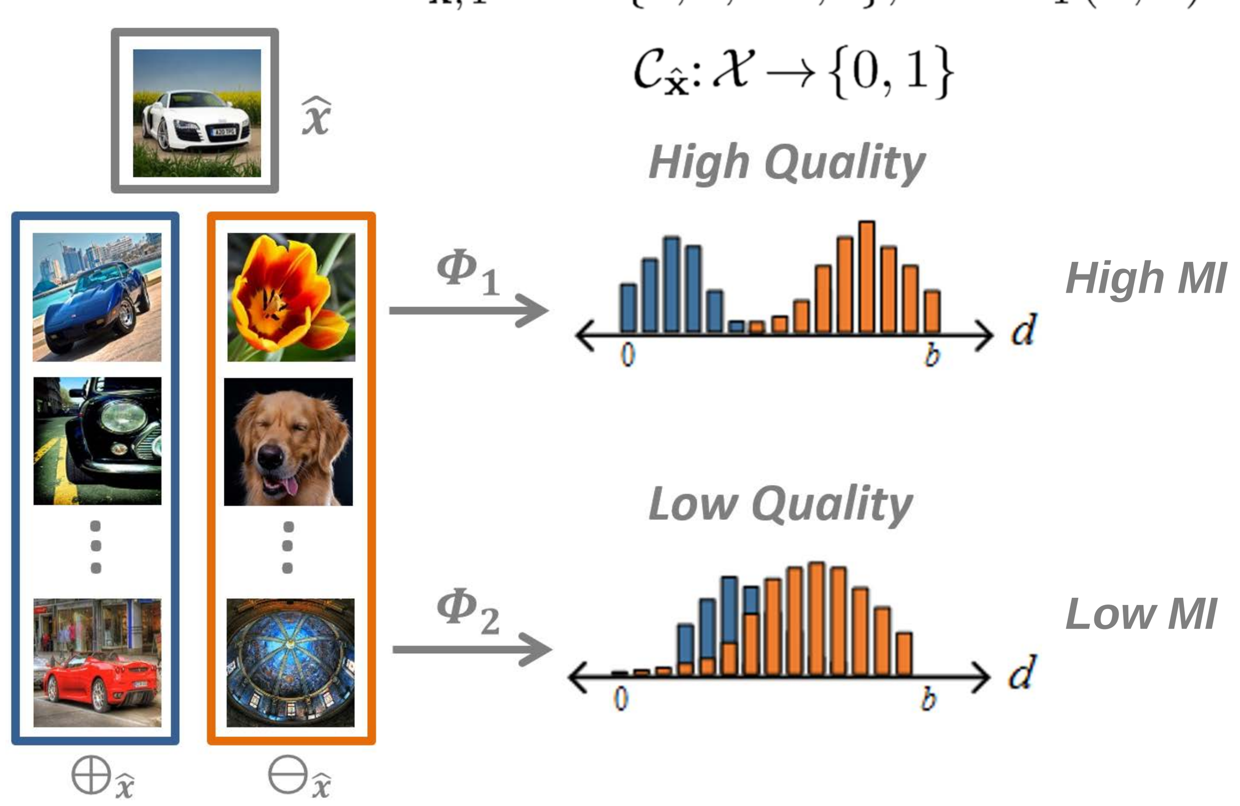
A: Only when hashing quality improves!

Quality Measure: Mutual Information

- **Mutual Information**: decrease in entropy of distance distribution, when conditioned on neighborhood info.

$$\mathcal{D}_{\hat{\mathbf{x}},\Phi}: \mathcal{X} \rightarrow \{0,1,\ldots,b\}, \mathbf{x} \rightarrow d_\Phi(\mathbf{x},\hat{\mathbf{x}})$$

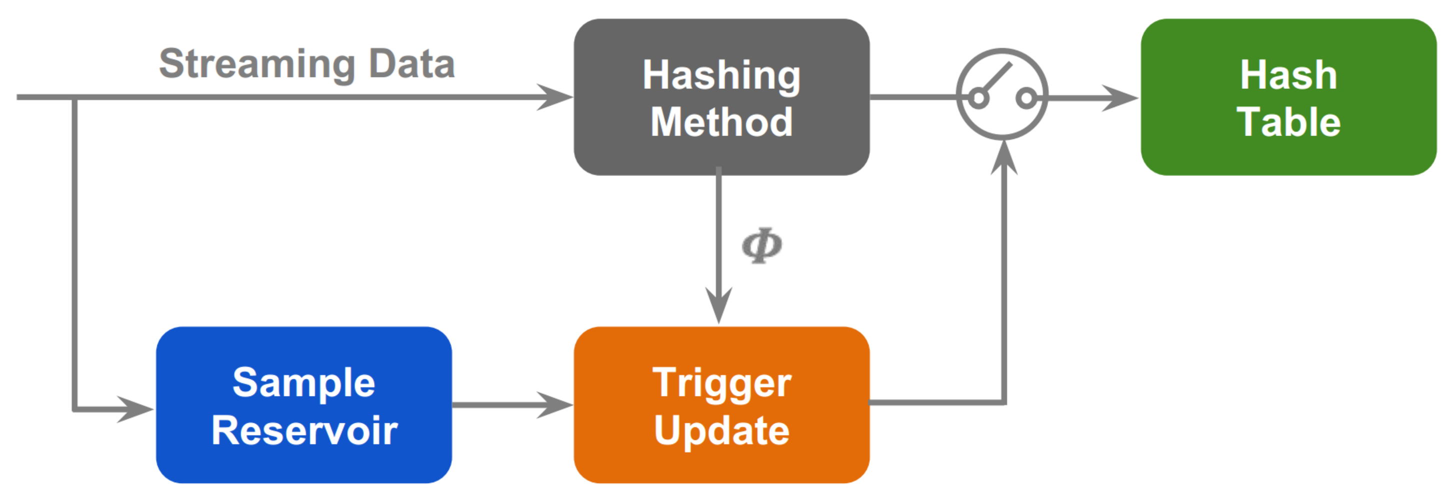
$$\mathcal{C}_{\hat{\mathbf{x}}}: \mathcal{X} \rightarrow \{0,1\}$$



- **Quality measure**: integrate over feature space

$$Q(\Phi) = \int_{\mathcal{X}} \mathcal{I}(\mathcal{D}_{\hat{\mathbf{x}},\Phi}; \mathcal{C}_{\hat{\mathbf{x}}}) p(\hat{\mathbf{x}}) d\hat{\mathbf{x}} = \int_{\mathcal{X}} (H(\mathcal{C}_{\hat{\mathbf{x}}}) - H(\mathcal{C}_{\hat{\mathbf{x}}}|\mathcal{D}_{\hat{\mathbf{x}},\Phi})) p(\hat{\mathbf{x}}) d\hat{\mathbf{x}}$$

Trigger Update: Plug-in Module



- Snapshot Φ^s, current Φ_t
- Update if

$$Q(\Phi_t) - Q(\Phi^s) > \theta$$

Monte-Carlo approximation of Q (sampling from stream):

- Reservoir Sampling [Vitter 1985]

$$Q_{\mathcal{R}}(\Phi) = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{x}^r \in \mathcal{R}} \mathcal{I}_{\mathcal{R}}(\mathcal{D}_{\mathbf{x}^r,\Phi}; \mathcal{C}_{\mathbf{x}^r})$$

MIHash: Gradient-based MI Optimization

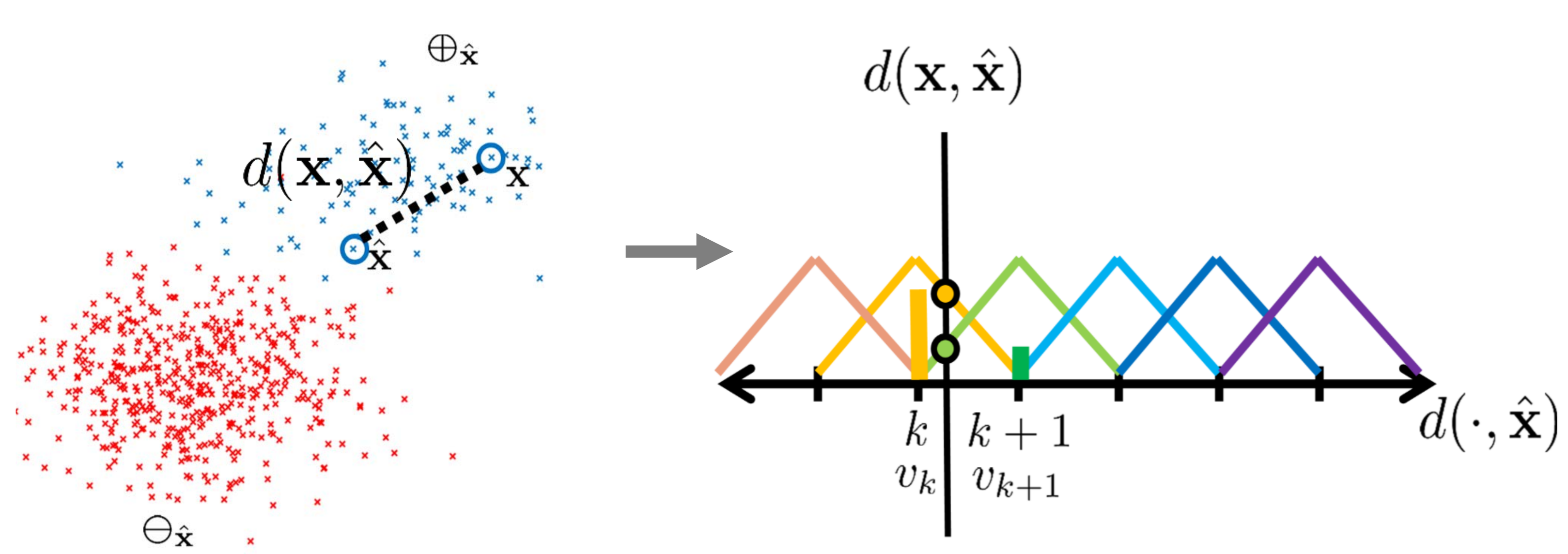
$$\frac{\partial \mathcal{I}(D_{\hat{\mathbf{x}},\Phi}; \mathcal{C}_{\hat{\mathbf{x}}})}{\partial \Phi(\hat{\mathbf{x}})} = \frac{\partial H(\mathcal{D}_{\hat{\mathbf{x}},\Phi})}{\partial \Phi(\hat{\mathbf{x}})} - \frac{\partial H(\mathcal{D}_{\hat{\mathbf{x}},\Phi}|\mathcal{C}_{\hat{\mathbf{x}}})}{\partial \Phi(\hat{\mathbf{x}})}$$

$$= \sum_l \frac{\partial H(\mathcal{D}_{\hat{\mathbf{x}},\Phi})}{\partial p_{\mathcal{D},l}} \frac{\partial p_{\mathcal{D},l}}{\partial \Phi(\hat{\mathbf{x}})}$$

$$- \left[p^+ \sum_l \frac{\partial H(\mathcal{D}_{\hat{\mathbf{x}},\Phi})}{\partial p^+_{\mathcal{D},l}} \frac{\partial p^+_{\mathcal{D},l}}{\partial \Phi(\hat{\mathbf{x}})} - p^- \sum_l \frac{\partial H(\mathcal{D}_{\hat{\mathbf{x}},\Phi})}{\partial p^-_{\mathcal{D},l}} \frac{\partial p^-_{\mathcal{D},l}}{\partial \Phi(\hat{\mathbf{x}})}\right]$$

Optimization approach: continuous relaxation

- Differentiable histogram binning [Usnitova & Lempitsky, NIPS'16]
- Binary bits: sigmoid approximation



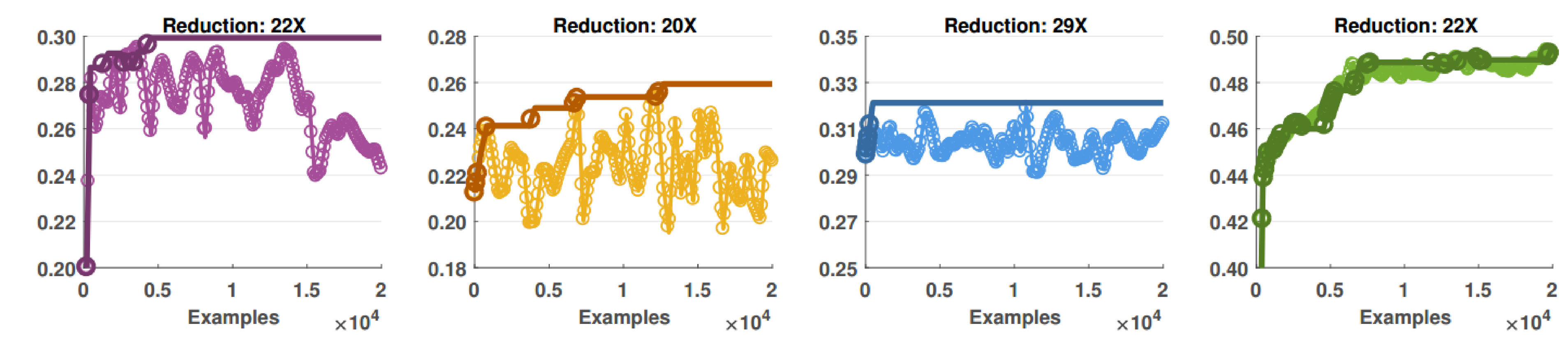
https://github.com/fcakir/mihash

Experiments: Online Hashing

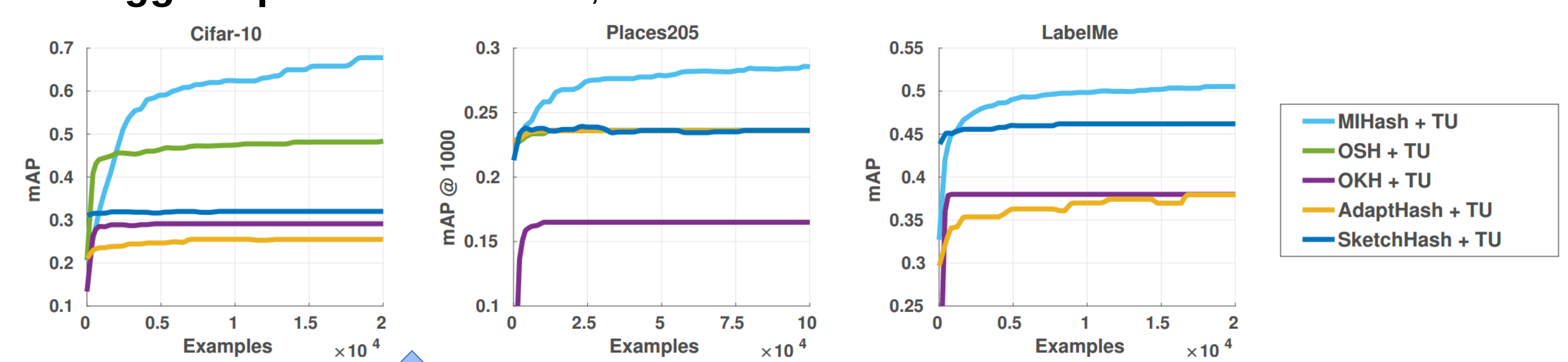
Datasets: CIFAR-10 (60K), Places205 (**2.5M**), LabelMe (22K)

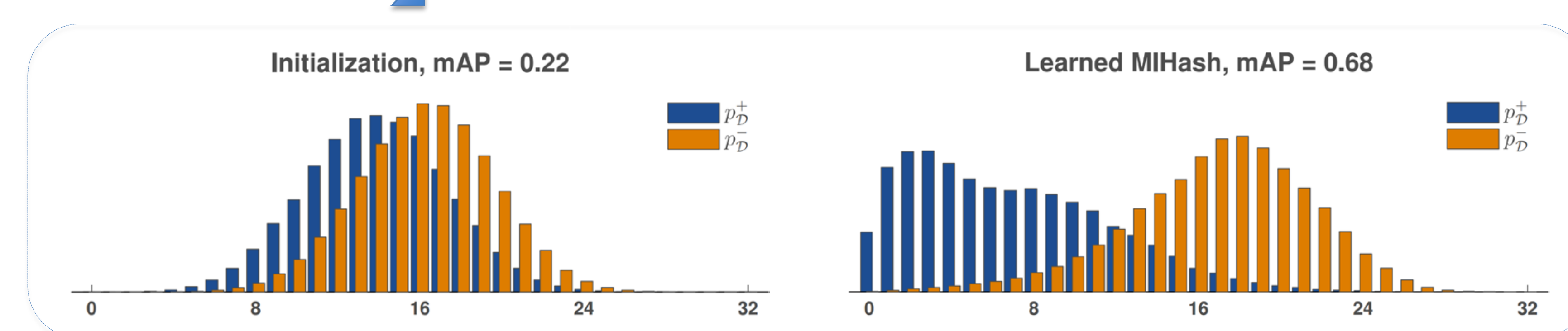
OKH [IJCAI'13], AdaptHash [ICCV'15], SketchHash [CVPR'15], OSH [ICIP'15]

Trigger Update on *vs.* off : CIFAR-10, 32 bits, mAP



Trigger Update on : 32 bits, mAP





Experiments: Batch Hashing

Batch Hashing: CIFAR-10, mAP

Setting 1: 5K / 1K, VGG-F fc7. Setting 2: 50K / 10K, VGG-F end-to-end.

Methods:

SHK [CVPR'12]

SDH [CVPR'15]

FastHash [CVPR'14]

VDSH [CVPR'16]

DPSH [IJCAI'16]

DTSH [ACCV'16]

		Code Length			
	Method	12	24	32	48
Setting 1	SHK	0.497	0.615	0.645	0.682
	SDH	0.521	0.576	0.589	0.592
	VDSH	0.523	0.546	0.537	0.554
	DPSH	0.420	0.518	0.538	0.553
	DTSH	0.617	0.659	0.689	0.702
	FastHash	0.632	0.700	0.724	0.738
	MIHash[1]	0.524	0.563	0.597	0.609
	MIHash	**0.683**	**0.720**	**0.727**	**0.746**
	Method	16	24	32	48
Setting 2	DPSH[2]	0.763	0.781	0.795	0.807
	DTSH[2]	0.915	0.923	0.925	0.926
	DPSH	0.908	0.909	0.917	0.932
	DTSH	0.916	0.924	0.927	0.934
	MIHash	**0.929**	**0.933**	**0.938**	**0.942**

[1] Trained for a single epoch. [2] Original reported results.