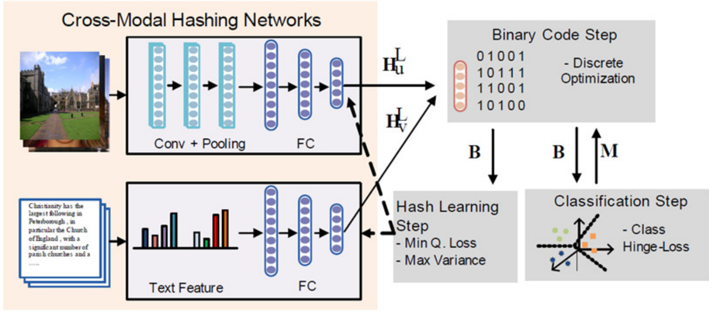
Learning a Cross-Modal Hashing Network for Multimedia Search

Venice Erin Liong[1,3], Jiwen Lu[2], Yap-Peng Tan[3]

[1]Interdisciplinary Graduate School (IGS) [3]School of Electrical and Electronic Engineering (EEE) Nanyang Technological University, Singapore
[2]School of Automation Tsinghua University, Beijing, China

Overview

- ✓ Learn compact binary codes for **cross-modality multimedia search**
- ✓ We design a deep neural network implementation:
  - ✓ Learns unified binary code discretely and discriminatively through a classification-based hinge-loss criterion
  - ✓ Cross-modal hashing network (CMHN), one deep network for each modality, through minimizing the quantization loss between real-valued neural code and binary code, and maximizing the variance of the learned neural codes



Formulation

❑ Problem:

$$f_u : \mathbb{R}^{d_u} \rightarrow \{-1,1\}^K, \quad f_v : \mathbb{R}^{d_v} \rightarrow \{-1,1\}^K$$

❑ The objective function of **CMHN**:

$$\min_{\mathbf{B},\mathbf{M},\theta_u,\theta_v} J \;=\; J_1 + \lambda_1 J_2$$

✓ **Binary codes (B)** by following the assumption that the codes should be able to perform well on a multi-classification problem (using hinge-loss criterion)

$$\min_{\mathbf{B},\mathbf{M}} J_1 \;=\; \|\mathbf{M}\|_F^2 + \sum_{n}^{N} \xi_n$$

$$\forall n, j \;\; \mathbf{y}_{n,j}(\mathbf{m}_j^\top \mathbf{b}_n) \geq 1 - \xi_n$$

✓ **Network Parameters θ_u, θ_v** by minimizing the quantization loss between neural codes and binary code and maximizes the variances

$$\min_{\theta_u,\theta_v} J_2 \;=\; (\|\mathbf{B}-\mathbf{H}_u^L\|_F^2 + \|\mathbf{B}-\mathbf{H}_v^L\|_F^2)$$
$$-\alpha\left(\mathrm{tr}(\mathbf{H}_u^L\mathbf{H}_u^{L\top}) + \mathrm{tr}(\mathbf{H}_v^L\mathbf{H}_v^{L\top})\right)$$

Optimization

We perform optimization by fixing the other variables and solving one variable alternatively and iteratively.

❑ Classification Step: learn the classification matrix (**M**) by having a support vector machine (SVM) formulation which can solved through a standard solver (libsvm)

❑ Binary Code Step: learn **B** by having a binary quadratic problem which can be solved through a linear gradient technique as follows:

$$\mathbf{b}_n \;=\; \mathrm{sgn}(\mathbf{y}_n\mathbf{M}^\top + \lambda_1(\mathbf{h}_{un}^L + \mathbf{h}_{vn}^L))$$

❑ Hash Function Learning Step: learn network parameters by a batch-wise gradient descent method.

❑ Implementation details:

- ❑ image network – pretrained CNN-F up to FC7 + new FC layer + hashlayer (tanh)
- ❑ Text network – FC layers [D – 500 - K]

❑ Out-of-sample extension:

$$\mathbf{b}_{un} = \mathrm{sgn}(\mathbf{h}_{un}^L), \mathbf{b}_{vn} = \mathrm{sgn}(\mathbf{h}_{vn}^L)$$

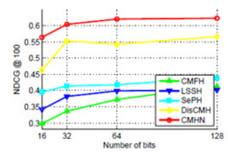
Experiments

❑ **IAPRTC12**: 19627 image-sentence pairs. Top 22 frequent labels from the 275 concept. Text feature:BoW 1386 dim

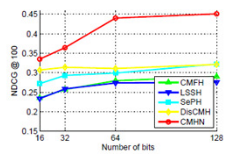
	Method	16 bits	32 bits	64 bits	128 bits
$I \rightarrow T$	CMFH [7]	0.5601	0.5829	0.6079	0.6179
	LSSH [6]	0.5440	0.5769	0.5964	0.5985
	SePH - *km* [10]	0.6177	0.6447	0.6500	0.6781
	DisCMH [12]	0.6174	0.6596	0.6503	0.6594
	DNH-C [26]	0.5250	0.5592	0.5902	0.6339
	DVSH [16]	0.5696	0.6321	0.6964	0.7236
	CMHN	**0.6483**	**0.7274**	**0.7974**	**0.8251**
	CMHN (o)	0.5768	0.7062	0.7780	0.8060
$T \rightarrow I$	CMFH [7]	0.5592	0.5834	0.6084	0.6187
	LSSH [6]	0.4868	0.5264	0.5547	0.5724
	SePH - *km* [10]	0.6105	0.6340	0.6404	0.6730
	DisCMH [12]	0.6532	0.6910	0.6921	0.6949
	DNH-C [26]	0.4692	0.4838	0.4905	0.5053
	DVSH [16]	0.6037	0.6395	0.6806	0.6751
	CMHN	**0.6687**	**0.6925**	**0.7535**	**0.7925**
	CMHN (o)	0.6716	0.6615	0.6677	0.6490

❑ **NUSWIDE**: 186577 images-tag pairs . Top 10 frequent concepts from 81 concepts. Text feature: BoW 1000 dim

	Method	16 bits	32 bits	64 bits	128 bits
$I \rightarrow T$	CMFH [7]	0.4772	0.5301	0.5763	0.6258
	LSSH [6]	0.5547	0.5734	0.5980	0.5968
	SePH - *km* [10]	0.6177	0.6447	0.6500	0.6781
	DisCMH [12]	0.6826	0.7583	0.7752	0.7605
	CAH [15]	0.4920	0.5084	0.5407	0.5628
	DCMH [17]	0.6249	0.6355	0.6720	-
	CMHN	**0.7893**	**0.8170**	**0.8236**	**0.8289**
	CMHN (o)	0.6558	0.7480	0.7818	0.7614
$T \rightarrow I$	CMFH [7]	0.4965	0.5432	0.5995	0.6405
	LSSH [6]	0.5857	0.6242	0.6293	0.6464
	SePH - *km* [10]	0.6604	0.6766	0.7043	0.7024
	DisCMH [12]	0.6519	0.7378	0.7535	0.7511
	CAH [15]	0.5019	0.5135	0.5451	0.5800
	DCMH [17]	0.6791	0.6829	0.6906	-
	CMHN	**0.6829**	**0.7469**	**0.7651**	**0.7772**
	CMHN (o)	0.6643	0.6950	0.7170	0.7062



NUSWIDE ($I \rightarrow T$)



(a) IAPRTC12 ($I \rightarrow T$)

Acknowledgements

This research was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the Infocomm Media Development Authority, Singapore