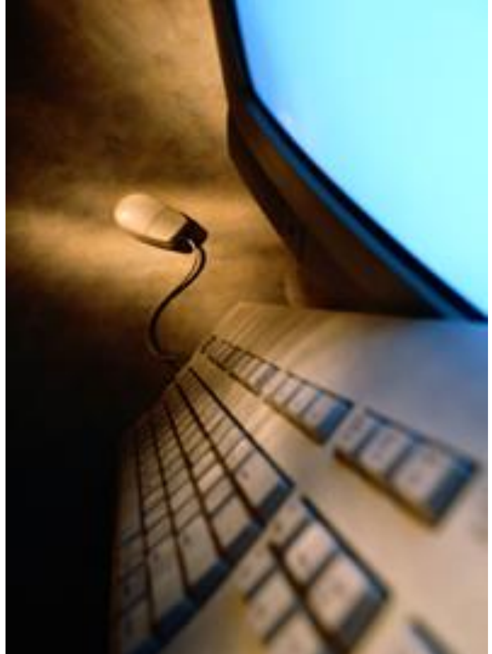
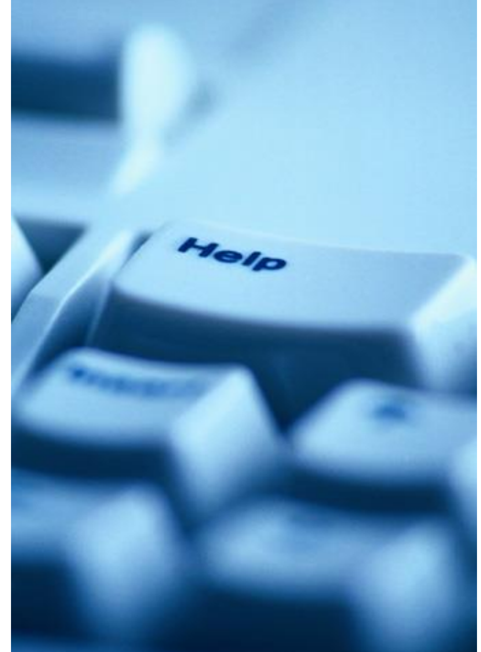
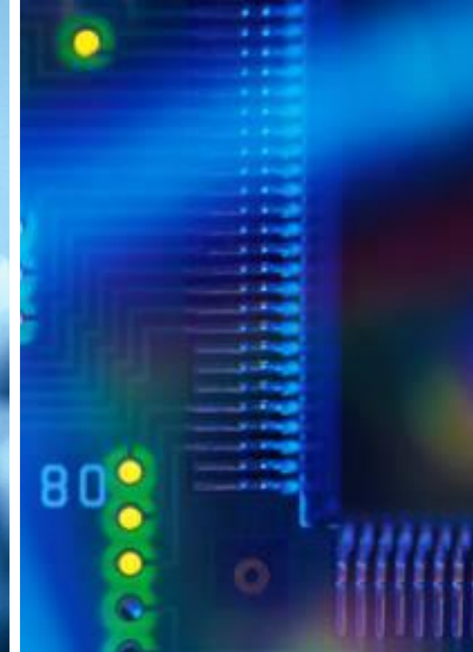
西南大学
SOUTHWEST UNIVERSITY
1906







Cycle-Consistent Deep Generative Hashing for Cross-Modal Retrieval

From 2019 TIP

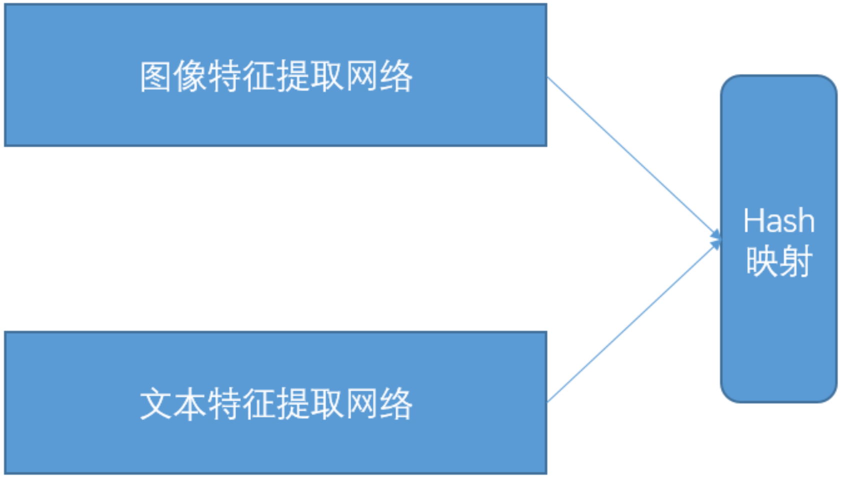
Lin Wu, Yang Wang and Ling Shao *Senior Member, IEEE*

By godder

背景介绍

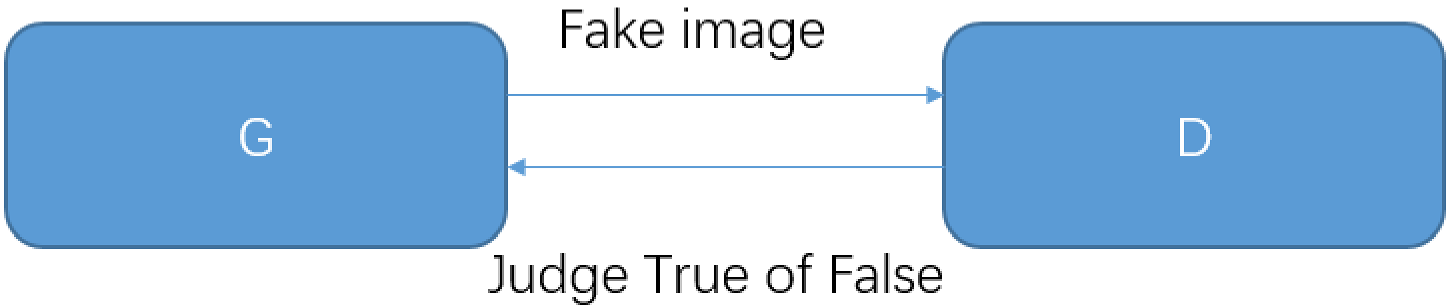
基于深度学习的跨模态hash方法：

1. 利用深度学习的方法训练针对不同模态的网络模型
2. 将不同模态的特征提取出相同的维度，然后进行hash映射
3. 将上述两部分利用全连接层连接，实现end-to-end的网络结构



背景介绍

GAN目前的应用与类型
1. 模型:



背景介绍

GAN目前的应用与类型
2. 目标函数:

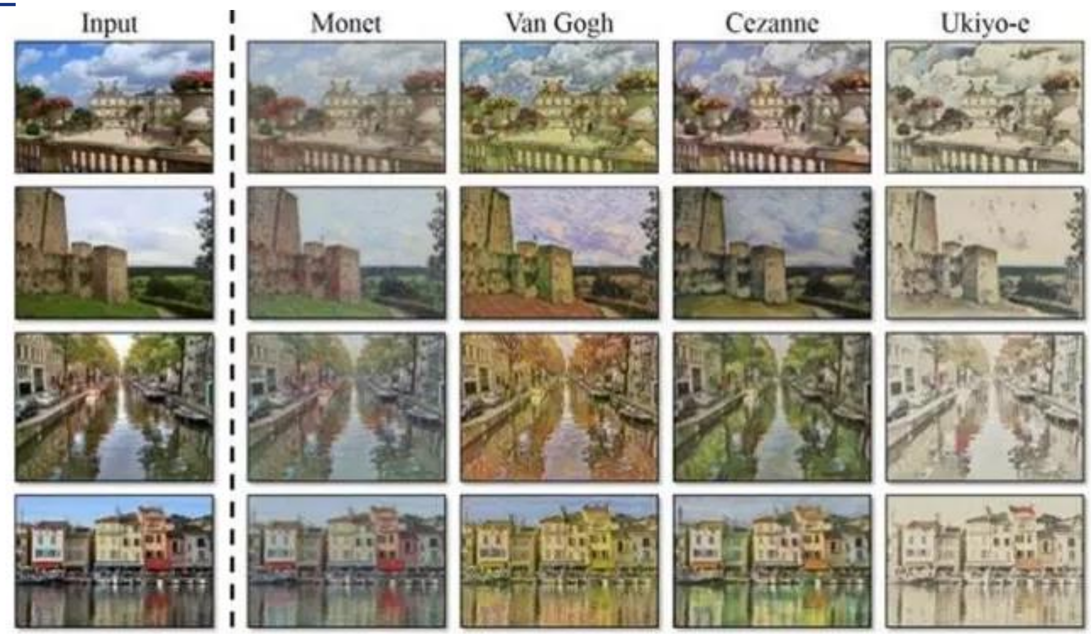
$$\min_G \max_D E_{x\sim q(x)}[\log D(x)] + E_{z\sim p(z)}[\log(1-D(G(z)))]$$

$$Loss\_D(L,D) = E_{x_r}(-\log(D(x_r))) + E_{x_f}(-log(1-D(x_f)))$$

$$Loss\_G = -\log D(x_f) = -(1*\log D(x_f) + 0*\log(1-D(x_f)) = \quad E_{xf}(log(1-D(xf)))$$

背景介绍

GAN目前的应用与类型
3. 应用:
风格迁移

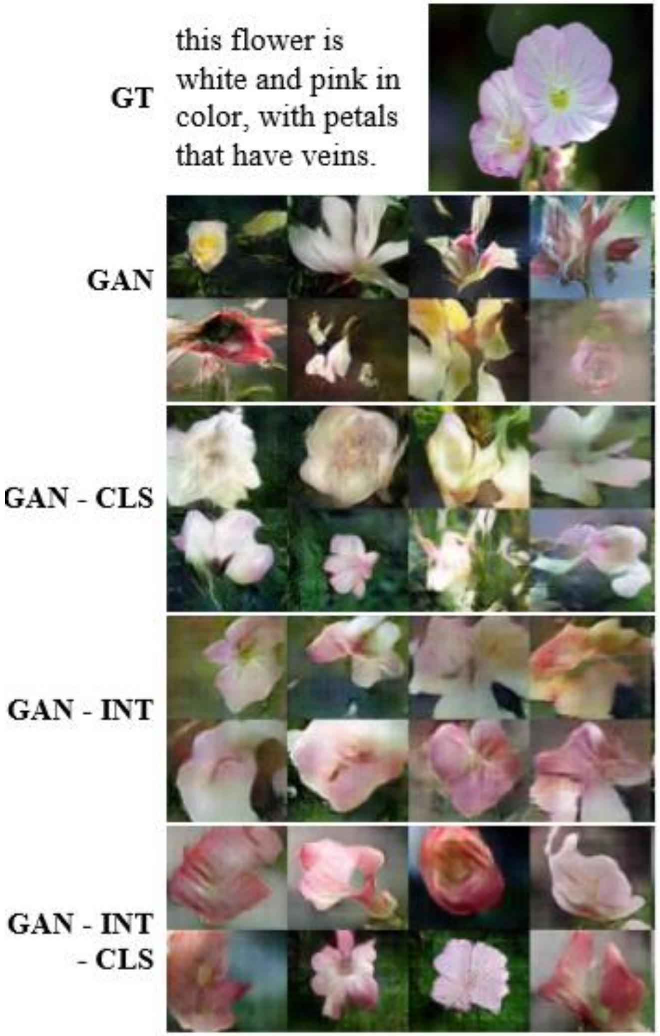


背景介绍

GAN目前的应用与类型
3. 应用：
风格迁移
图像增强

背景介绍

GAN目前的应用与类型
3. 应用：
风格迁移
图像增强
跨模态对象生成



背景介绍

GAN目前的应用与类型
4. 存在问题：
模型坍塌问题
训练难

背景介绍

GAN目前的应用与类型
5. 现有模型：

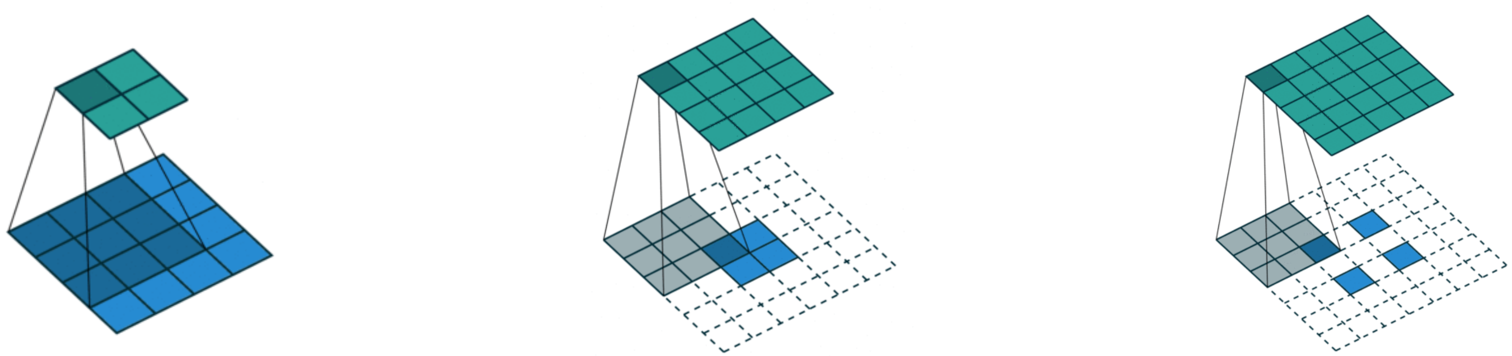
常见模型：
Dcgan
Cgan
Pix2pix
Cyclegan

aae
acgan
began
bgan
bicyclegan
ccgan
cgan
cogan
context_encoder
cyclegan
dcgan
discogan
dragan
dualgan
ebgan
gan
infogan
lsgan
munit
pix2pix
pixelda
sgan
softmax_gan
srgan
stargan
unit
wgan
wgan_div
wgan_gp

相关概念及数学公式

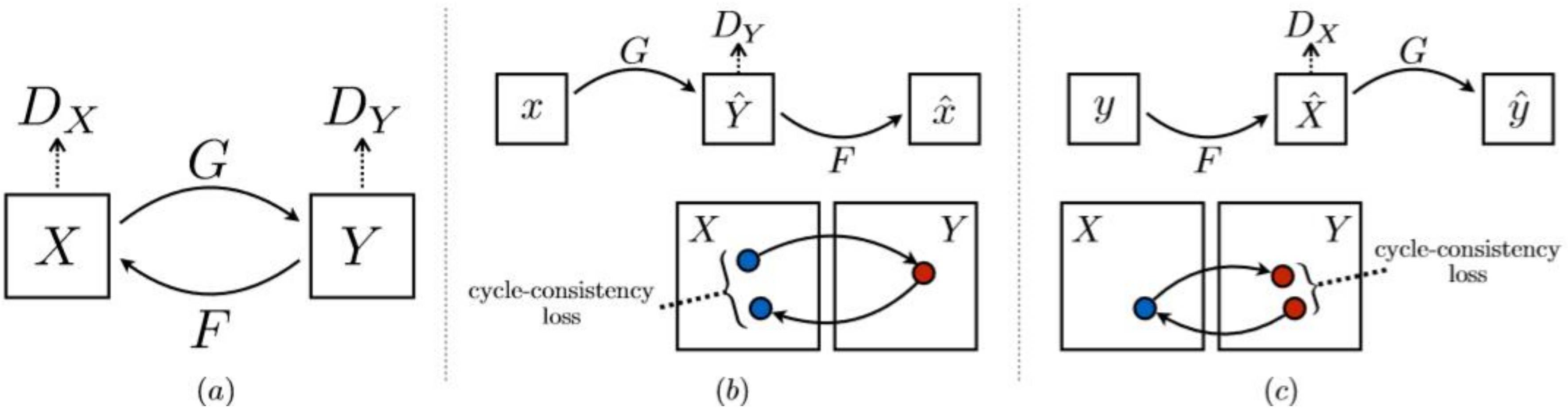
1. CNN

Convolution neural network（卷积神经网络）通过卷积核对输入中指定大小的矩阵进行相应位置的乘法，并将结果加和输出。按照一般理解，卷积其实就是一种滤波，通过卷积滤波后的图像中对应于该卷积核的特征会突出显示，配合卷积之后的一般是pooling（池化）操作，用来将通过卷积后获得的突出特征进行筛选，剔除非强调特征。随着卷积与池化的结合，将一个图像的浅层语义到深层语义依次筛选出。同时卷积包括其不同类型，如转置卷积(反卷积)、微步卷积。如下图分别为卷积操作、反卷积操作、微步卷积操作



相关概念及数学公式

2. Cycle GAN



相关概念及数学公式

2. Cycle GAN

$$\mathcal{L}(G,F,D_X,D_Y) = \mathcal{L}_{\text{GAN}}(G,D_Y,X,Y) + \mathcal{L}_{\text{GAN}}(F,D_X,Y,X) + \lambda\mathcal{L}_{\text{cyc}}(G,F),$$

$$\mathcal{L}_{\text{GAN}}(G,D_Y,X,Y) = \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log(1-D_Y(G(x))], \quad (1)$$

$$\mathcal{L}_{\text{cyc}}(G,F) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\|F(G(x))-x\|_1] + \mathbb{E}_{y\sim p_{\text{data}}(y)}[\|G(F(y))-y\|_1]. \quad (2)$$

https://www.youtube.com/watch?v=AxrKVfjSBiA

Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks


Help


相关概念及数学公式

3. 信息熵

$$I(x) = \log \frac{1}{P(x)}$$

$$H(P) = \sum_x P(x) \log \frac{1}{P(x)}$$

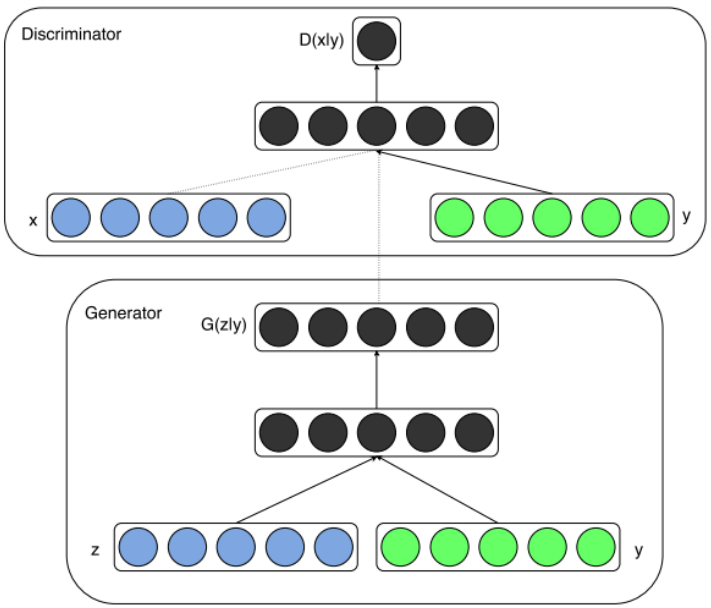
$$H_P(Q) = \int_x Q(x) \log \frac{1}{P(x)} \, dx$$

$$D_P(Q) = H_p(Q) - H(Q)$$

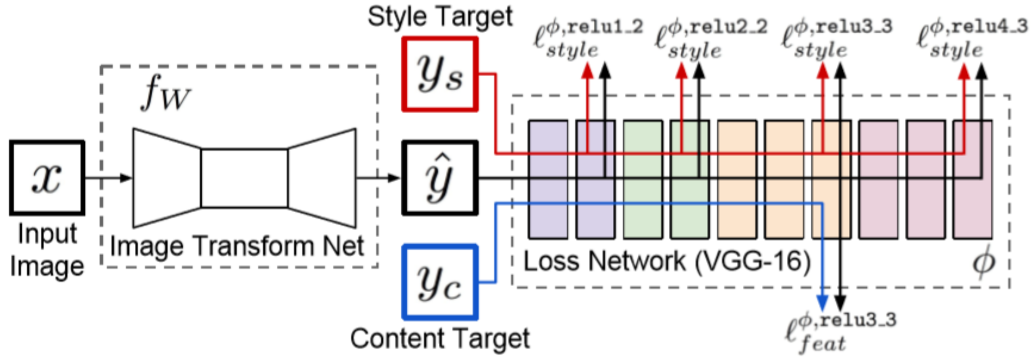
相关概念及数学公式

4. 扩充内容

C GAN



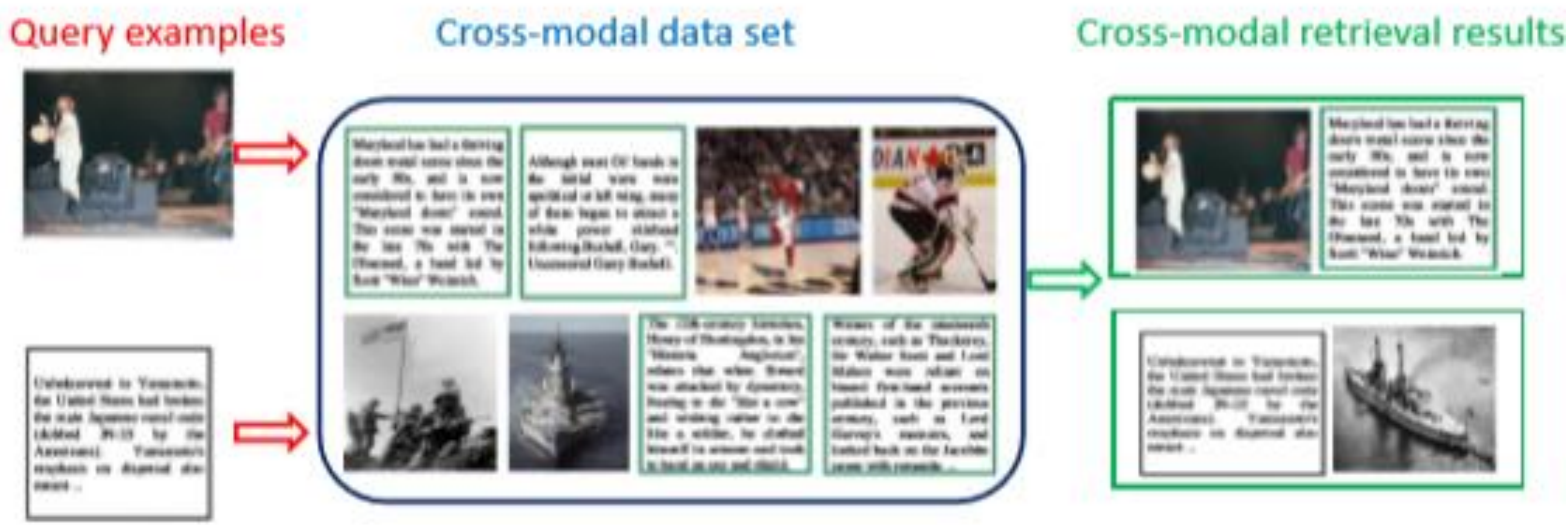
pix2pix



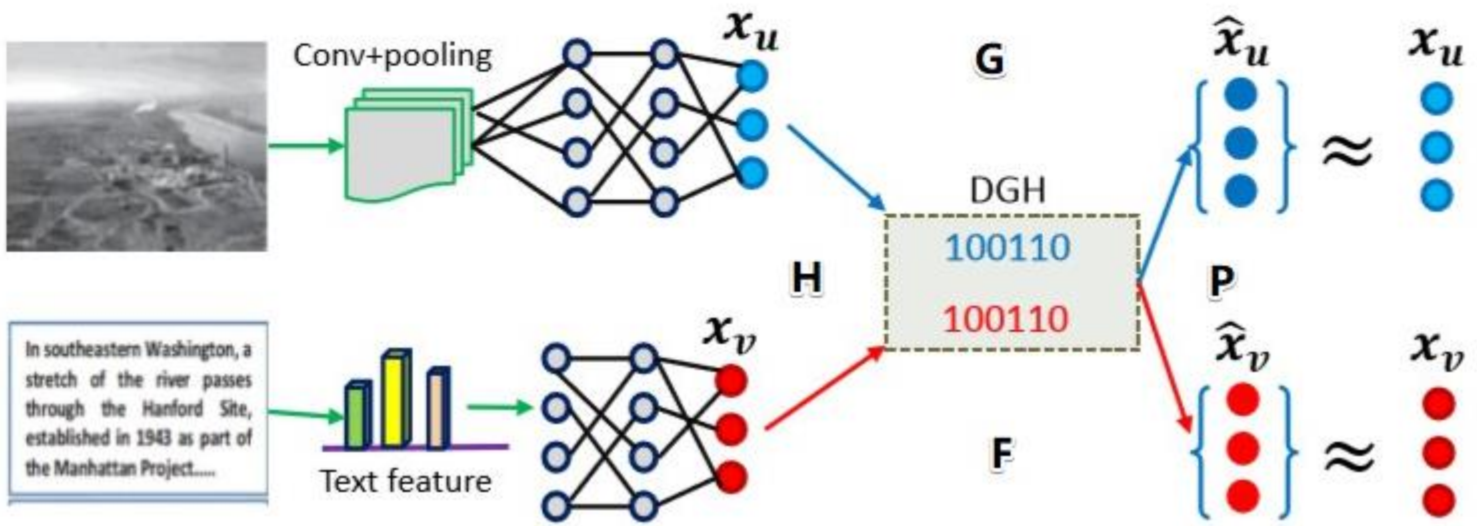
Perceptual Losses for Real-Time Style Transfer and Super-Resolution

Conditional Generative Adversarial Nets

introduction



Network structure



Cycle-Consistent Deep Generative Hashing (CYC-DGH)

Network structure

Txt Encoding Net

Dataset	Layer	config
coco	FC1	1000
	FC2	500
	FC3	200
IAPR TC-12	FC1	11500
	FC2	500
	FC3	200
Wiki	FC1	10
	FC2	500
	FC3	200
	FC1	128 with leaky relu

Generative Adversarial Text to Image Synthesis

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran REEDSCOT[1], AKATA[2], XCYAN[1], LLAJAN[1]
Bernt Schiele, Honglak Lee SCHIELE[2],HONGLAK[1]
[1] University of Michigan, Ann Arbor, MI, USA (UMICH.EDU)
[2] Max Planck Institute for Informatics, Saarbrücken, Germany (MPI-INF.MPG.DE)

Network structure

Image Generator Net

Layer	config
Conv1	Kernel = 9, stride=1, out=32
Conv2	Kernel = 3, stride=2, out=64
Conv3	Kernel = 3, stride=2, out=128
Residual Block several	Kernel = 3, stride=1, out=128, num=5
Deconv1	Kernel = 3, stride=1/2, out=64
Deconv2	Kernel = 3, stride=1/2, out=32
Deconv3	Kernel = 3, stride=1, out=3

Perceptual Losses for Real-Time Style Transfer and Super-Resolution

Justin Johnson, Alexandre Alahi, and Li Fei-Fei

Department of Computer Science, Stanford University
{jcjohns, alahi, feifeili}@cs.stanford.edu

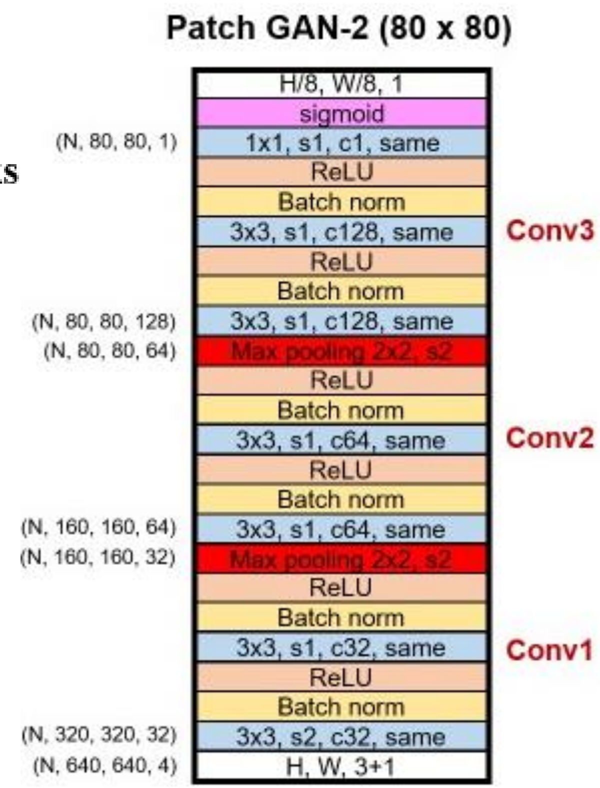
Network structure

Image Discriminator Net

Image-to-Image Translation with Conditional Adversarial Networks

Phillip Isola Jun-Yan Zhu Tinghui Zhou Alexei A. Efros

Berkeley AI Research (BAIR) Laboratory, UC Berkeley



Loss and objective

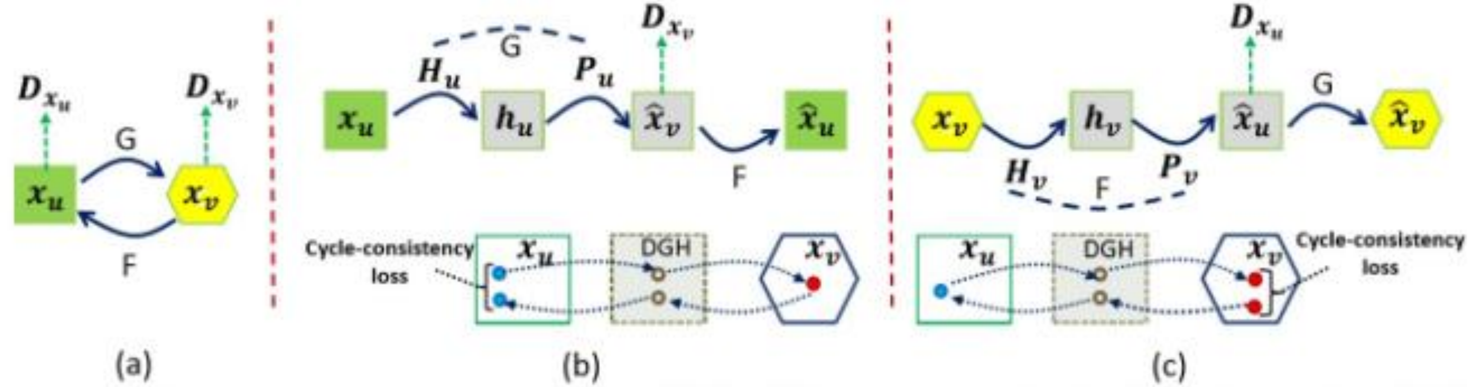


Fig. 3: The proposed cycle-consistent deep generative hashing (CYC-DGH) for cross-modal retrieval. (a) The model of CYC-DGH couples two mappings: $\boldsymbol{G}: \boldsymbol{x}_u \rightarrow \boldsymbol{x}_v$ and $\boldsymbol{F}: \boldsymbol{x}_v \rightarrow \boldsymbol{x}_u$ as well as associated adversarial discriminators $\boldsymbol{D}_{\boldsymbol{x}_v}$ and $\boldsymbol{D}_{\boldsymbol{x}_u}$. The two mappings are decomposed into the binary code generation and the reverse process of regenerating inputs from binary codes: $\boldsymbol{G}: \boldsymbol{x}_u \rightarrow \boldsymbol{H}_u \rightarrow \boldsymbol{P}_u \rightarrow \boldsymbol{x}_v$ and $\boldsymbol{F}: \boldsymbol{x}_v \rightarrow \boldsymbol{H}_v \rightarrow \boldsymbol{P}_v \rightarrow \boldsymbol{x}_u$. To regulate the mappings, two cycle-consistent losses are introduced: (b) forward $\boldsymbol{x}_u \rightarrow G(\boldsymbol{x}_u) \rightarrow \boldsymbol{F}(\boldsymbol{G}(\boldsymbol{x}_u)) \approx \hat{\boldsymbol{x}}_u$, and (c) backward $\boldsymbol{x}_v \rightarrow F(\boldsymbol{x}_v) \rightarrow \boldsymbol{G}(\boldsymbol{F}(\boldsymbol{x}_v)) \approx \hat{\boldsymbol{x}}_v$.

Loss and objective

Adversarial loss

$$L_{GAN}(G, D_{x_v}, x_u, x_v) = E_{x_v \sim p_{data}(x_v)}[logD_{x_v}(x_v)] + E_{x_u \sim p_{data}(x_u)}[log(1 - D_{x_v}(G(x_u)))]$$

$$L_{GAN}(G, D_{x_u}, x_v, x_u) = E_{x_u \sim p_{data}(x_u)}[logD_{x_u}(x_u)] + E_{x_v \sim p_{data}(x_v)}[log(1 - D_{x_u}(G(x_v)))]$$

这里与一般的GAN不同的是，loss并没有采用nll loss，而是采用了least-square loss（最小二乘法loss），

$$train\ for\ generator\ to\ minimize: E_{x_v \sim p_{data}(x_v)}[(D_{x_v}(x_v) - 1)^2]$$

$$train\ for\ distinguish\ to\ minimize: E_{x_u \sim p_{data}(x_u)}[(D_{x_v}(x_u) - 1)^2] + : E_{x_v \sim p_{data}(x_v)}[D_x(x_v)]$$

Loss and objective

cycle-consistency loss

$$L_{cyc}(G, F) = E_{x_v \sim p_{data}(x_u)}[||F(G(x_u)) - x_u||1] + Ex_u \sim p_{data}(x_u)[||G(F(x_v)) - x_u||_1]$$

Loss and objective

deep generative hashing

1. hash 生成 feature map

我们先做如下定义:

$P_u : h_u \to x_v$， $denoted\ as\ p(x_v|h_u)$　　$P_v : h_v \to x_u$， $denoted\ as\ p(x_u|h_v)$

We use a simple Gaussian distribution to model the generation of x given h:

$$p(x,h)=p(x|h)p(h),\ \text{where}\ p(x|h)=\mathcal{N}(Uh,\rho^2 I)$$

Loss and objective

deep generative hashing

stochastic generative hashing

1. hash 生成 feature map

我们先做如下定义:

$P_u : h_u \to x_v$， $denoted\ as\ p(x_v|h_u)$ $P_v : h_v \to x_u$， $denoted\ as\ p(x_u|h_v)$

We use a simple Gaussian distribution to model the generation of x given h:

$$p(x,h) = p(x|h)p(h), \text{ where } p(x|h) = \mathcal{N}(Uh, \rho^2 I)$$

$$p(x,h) \propto \exp\left(\frac{1}{2\rho^2}\underbrace{\left(x^\top x + h^\top U^\top U h - 2x^\top U h\right)}_{\|x - U^\top h\|_2^2} - \left(\log\frac{\theta}{1-\theta}\right)^\top h\right)$$

其中高斯重构误差为$\|x-U^T h\|_2^2$表示欧式领域稳定程度，当范数U是有限的时候，误差越小表示稳定性越高

Loss and objective

deep generative hashing

2.Feature map 生成 hash

由目前再自动编码上的研究，在概率模型p(h|x)上寻找最优解是很难的，所以这里依旧借助SGH里的内容进行定义

$$q(h|x)=\prod_{k=1}^{l} q(h_k=1|x)^{h_k} q(h_k=0|x)^{1-h_k},$$

$$q(h|x)=\prod_{k=1}^{K} q(h_k=1|x)^{h_k} q(h_k=0|x)^{1-h_k}$$

其中$h=[h_k]k=1^K \sim B(\sigma(W^Tx))$是线性参数化的，其中$W=[w_k]k=1^K$

然后结合W进行优化，后得到优化后的结果

$$p(h|x)=arg\max_{h} q(h|x)=\frac{sign(W^Tx)+1}{2}$$

Loss and objective

Training objective

$$L(G, F, D_{x_u}, D_{x_v}, H) = L_{GAN}(G, D_{X_v}, x_u, x_v) + L_{GAN}(G, D_{x_u}, x_v, x_u) + \lambda L_{cyc}(G, F) + D_{KL}(q(h|x)||p(h|x)) + L(\theta; x_*)$$

其中 $= u, v, D_{KL}(p||q) = \sum_{x \in X}[p(x)logp(x) - p(x)logq(x)], L(\theta; x) = E_{q(h|x_*)}[-logq(h|x) + logp(x|h)]$ 其中 $\theta = W, U, \rho, \beta_* := log\frac{\theta}{1-\theta}$

Loss and objective

Training objective

由于目标函数关于p(x|h)得导数难以获得，所以本文依旧利用了SGH中得内容，将上式中得h进行替换，内容如下

定义 $$\tilde{h}(z,\xi) := \begin{cases} 1 & \text{if } z \geqslant \xi \\ 0 & \text{if } z < \xi \end{cases}.$$

$$\tilde{H}(\Theta) = \sum_x \tilde{H}(\Theta; x) := \sum_x \mathbb{E}_\xi \left[\ell(\tilde{h}, x) \right], \tag{6}$$

where $\ell(\tilde{h}, x) := -\log p(x, \tilde{h}(\sigma(W^\top x), \xi)) + \log q(\tilde{h}(\sigma(W^\top x), \xi)|x)$ with $\xi \sim \mathcal{U}(0,1)$. With such a reformulation, the new objective can now be optimized by exploiting the distributional stochastic gradient descent, which will be explained in the next section.

Training

本文在训练阶段中，所有经过辨别器D的数据都是之前生成的数据(可靠性高)，不是新生成的数据，这样能够保证网络的稳定性。同时超参数λ=10，学习率从前100epoch的0.0002动态降低至0

Experiments

- Datasets:
  (1) COCO
  (2) Wiki
  (3) IAPR TC-12
- Evaluation Criteria:
  (1) L2 reconstruction error
  (2) training time
  (3) mAP
  (4) Precision-Recall curve
- control experiment（对照试验）

- Baselines:
  (1) TUCH
  (2) CMDVH
  (3) DVSH
  (4) CorrAE
  (5) CMNN
  (6) CAH
  (7) DCMH
  (8) HashGAN

control experiment

本文利用对照试验，测试了cycle GAN, hash与feature map相互生成的loss的作用效果，测试结果以精度为准

Loss	Per-class accuracy	Per-pixel accuracy
Cycle alone	0.270	0.724
GAN alone	0.611	0.126
CYC-DGH	**0.584**	**0.192**

Performance of training time

Training time on Microsoft-COCO in seconds				
Method	16 bits	32 bits	64 bits	128 bits
CYC-DGH	4.23	6.38	9.71	12.35
ITQ [58]	22.74	38.36	51.91	67.23

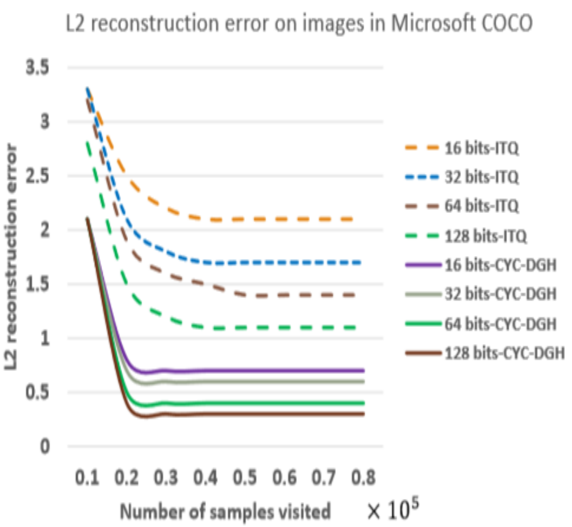
TABLE II: Training time comparison on Microsoft-COCO.

Training time on IAPR TC-12 in seconds				
Method	16 bits	32 bits	64 bits	128 bits
CYC-DGH	3.92	5.84	9.11	11.05
ITQ [58]	17.49	30.17	46.77	60.22

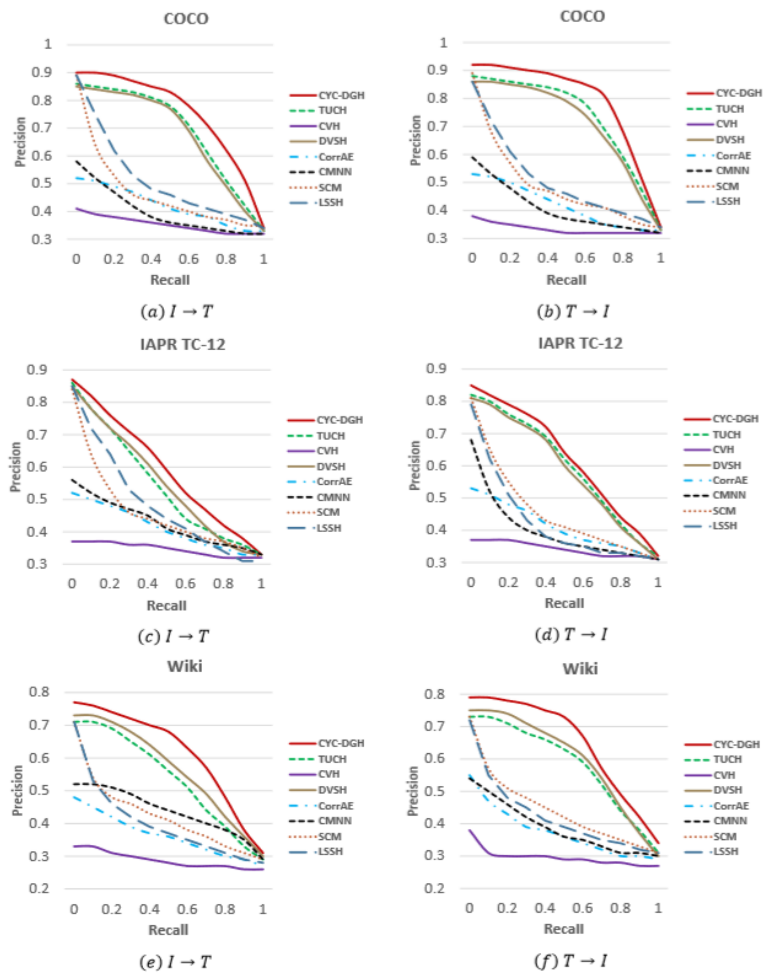
TABLE III: Training time comparison on IAPR TC-12.

control experiment

L2 reconstruction error



Performance of Precision-Recall Curve



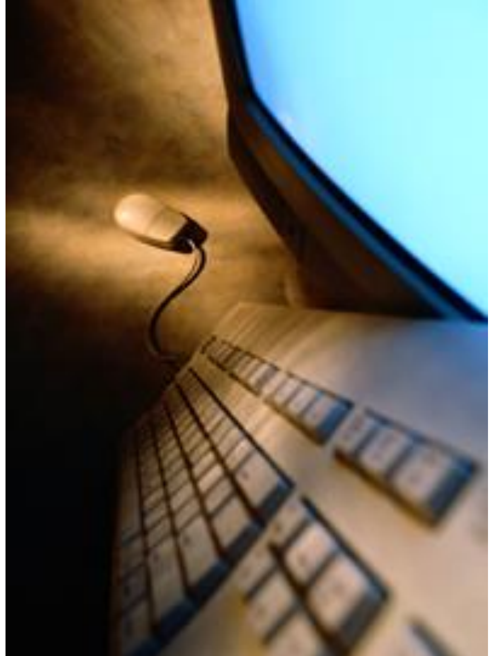
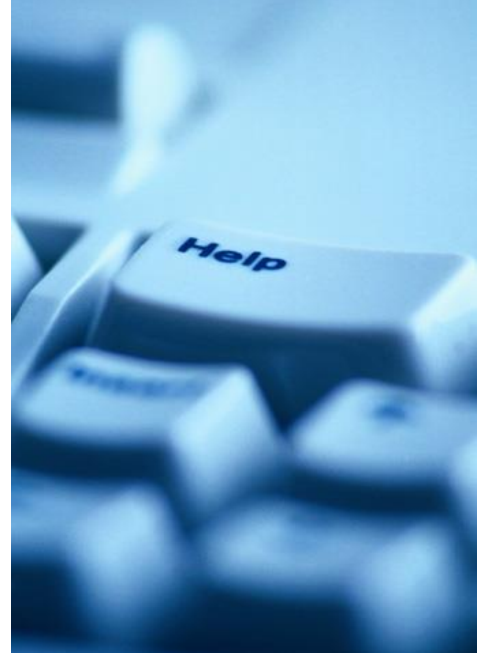
Performance of mAP

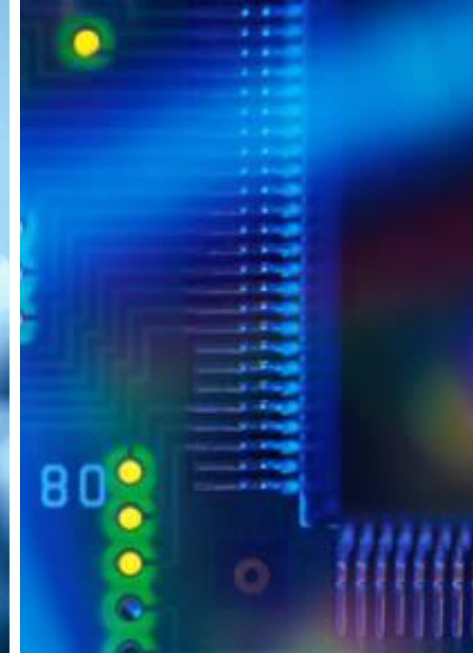
TABLE V: Mean Average Precision (MAP) comparison of state-of-the-art cross-modal hashing methods on three data sets.

Task	Method	Microsoft COCO				IAPR TC-12				Wiki			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
$I \rightarrow T$	CVH [35]	0.373	0.368	0.366	0.357	0.537	0.541	0.524	0.496	0.238	0.204	0.179	0.158
	SCM [19]	0.570	0.600	0.631	0.649	0.567	0.505	0.454	0.418	0.139	0.137	0.141	0.136
	LSSH [36]	-	-	-	-	0.544	0.577	0.596	0.599	0.364	0.371	0.378	0.358
	SePH [1]	0.581	0.613	0.625	0.634	0.618	0.645	0.650	0.678	0.414	0.435	0.437	0.447
	CYC-DGH	**0.722**	**0.754**	**0.781**	**0.780**	**0.771**	**0.815**	**0.832**	**0.831**	**0.794**	**0.811**	**0.813**	**0.820**
$T \rightarrow I$	CVH [35]	0.373	0.369	0.365	0.371	0.568	0.578	0.561	0.536	0.388	0.336	0.257	0.230
	SCM [19]	0.558	0.619	0.658	0.686	0.652	0.570	0.478	0.421	0.132	0.143	0.156	0.149
	LSSH [36]	-	-	-	-	0.487	0.526	0.555	0.572	0.606	0.626	0.638	0.638
	SePH [1]	0.613	0.650	0.672	0.693	0.610	0.634	0.640	0.673	0.701	0.699	0.710	0.715
	CYC-DGH	**0.761**	**0.796**	**0.834**	**0.859**	**0.772**	**0.798**	**0.837**	**0.842**	**0.811**	**0.823**	**0.826**	**0.822**

conclusion

本文利用了cycle gan并利用在跨模态上，同时结合SGH为hash与模态特征之间提供数学基础，利用cycle gan的特性靠近不同模态间的特征距离以及利用hash与特征的互相生成，靠近hash与特征的距离。



谢谢！