基于条件对抗网络的图像到图像转换

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

摘要

我们研究将条件对抗网络作为图像-图像转换的通用解决方案。条件对抗网络不仅能学习从输入图像到输出图像的映射，而且能学习一个用于训练输入-输出映射的损失函数。这让同一个通用的方案应用于传统的需要设计不同的损失函数的任务成为可能。我们证明这种方法在图像标签合成图像、边缘图重建物体图像、图像上色等问题上非常有效。自从与我们论文的开源代码 pix2pix 发布以来，许多互联网用户（包括许多艺术家）将我们的模型应用到工作中，更加证明了这个方法的泛用性和无需参数调整的易用性。使用深度学习方法后我们不再需要手动建立映射函数，这篇文章证明我们不用手动建立损失函数同样可以得到非常优秀的结果。

1. 简介

图像处理、计算机图形学和计算机视觉领域的许多问题都可描述为从输入图像转换成输出图像。像同一个概念可以使用英语和法语表达一样，场景也可以用 RGB 图像、图像梯度、边缘图或语义标签图的形式呈现。模仿自动语言翻译，我们将自动图像-图像转换定义为给定充足的训练数据将场景的某种表现形式翻译成另一种表现形式的任务。从目前已有的工作来看，上述几个任务中的每一个都已经被单独的、专用的机制很好地解决[16, 25, 20, 9, 11, 53, 33, 39, 18, 58, 62]，即使其目的总是从输入像素转换到输出像素。我们这篇文章的目的是为上述所有问题的解决提出一个通用框架。

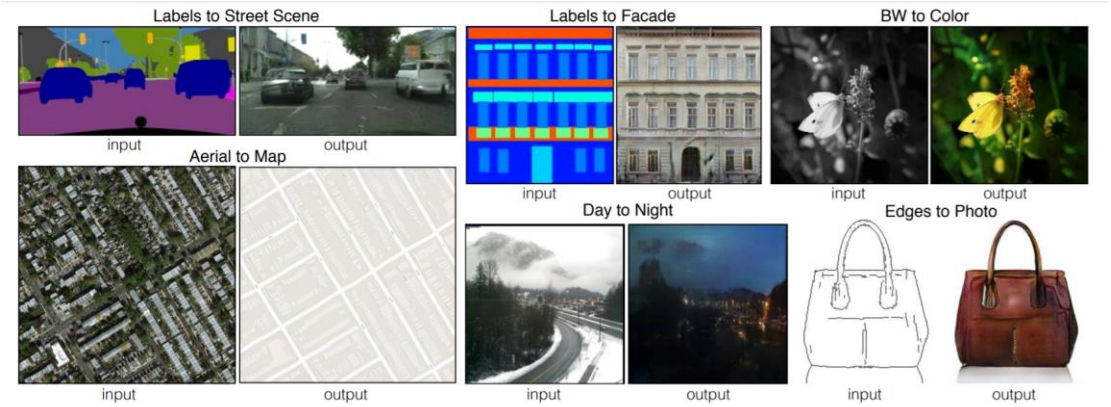


图 1 在图像处理、图形学和视觉相关的任务中许多任务需要将输入图像转换为对应的输出图像，这些任务常常使用针对任务具体化应用的算法，即使任务本质几乎相同：将像素映射到像素。条件对抗网络是在一系列上述任务中表现非常好的通用解决方案。这里我们展示了在几个任务上的结果。每个案例中我们都是用相同的结构和目标函数，只是使用不同的数据集训练。

社区已经在这个方向上走出了意义非常重大的一步，即使用卷积神经网络（CNNs）作为许多图像预测的通用解决办法。CNN 学习最小化度量结果的目标损失函数，尽管这个过程是自动的，还是需要投入许多人力努力设计有效的损失函数。但是就想麦达斯国王一样，我们一定要注意我们想要的是什么。如果使用比较传统一点的做法，让 CNN 最小化预测值和真实值之间的欧式距离，CNN 将会倾向于生成模糊的结果[43,61]。这是因为欧氏距离是通过让所有合理输出取平均值来实现最小化，从而导致生成结果模糊。如何让 CNN 输出我们真正想要的清晰的、真实的图片是一个热点问题，通常需要很多专业知识。

我们非常希望我们能够指出一个高级别的目标，比如让输出和真实图片难以区分，之后 CNN 能够自动学习满足目标的合适的损失函数。幸运的是，这正是最近提出的生成对抗网络（Generative Adversarial Networks ,GANs）所做的工作[24, 13, 44, 52, 63]。GANs 学习一个试图区分输出图片是真实的还是伪造的损失函数，同时训练一个最小化这个损失的生成模型，在这个模型中将不再生成模糊的结果因为它看起来明显是假的。因为 GANs 学习一个适用于数据的损失函数，他们可以被用于许多按照传统的做法需要精心设计损失函数的任务中。

在这篇文章中，我们探索 GANs 在条件设置方面的应用。仅仅因为 GANs 学习数据的

生成模型，条件 GANs（conditional Generative Adversarial Networks, cGANs）学习条件生成模型。这使 cGANs 非常适合图像-图像转换任务，我们只需要给出输入条件就能生成相应的输出图像。

在过去两年里，已经有许多关于 GANs 的研究，这篇文章中探索的许多技术都是前人已经使用过的。尽管如此，几乎此前的所有论文都将精力集中在具体应用上，使用 cGANs 作为图像-图像转换的通用解决办法效果如何还是未知数。我们的主要贡献是证明 cGANs 在许多图像-图像转换任务上表现非常好。我们的第二点贡献是提出了一个通用的、性能高的简单框架，并且分析了几个重要的结构选择的效果。项目源码见 https://github.com/phillipi/pix2pix。

2. 相关工作

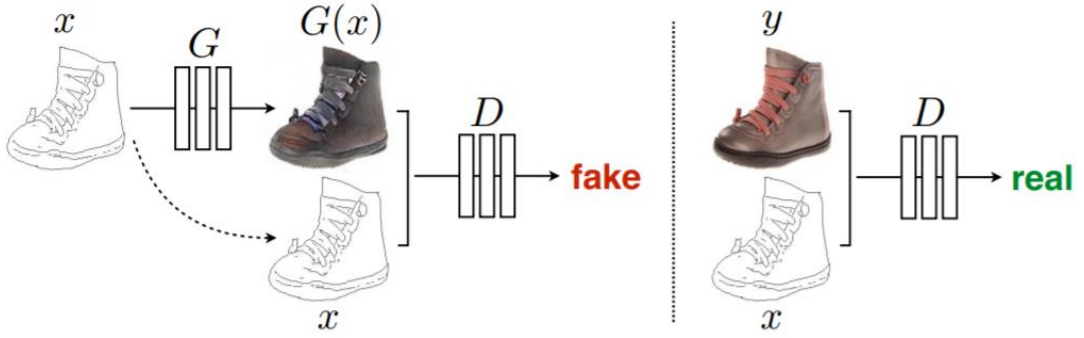


图 2 训练一个条件对抗网络完成边缘-图像转换任务。判别器 D 学习区分假图（生成器合成）和真实的图像。生成器 G 学习欺骗判别器。与非条件对抗网络不同，这里生成器和判别器都观察到了输入边缘图。

图像建模的结构化损失 图像-图像转换任务经常使用逐个像素分类或回归任务公式描述[39, 58, 28, 35, 62]。这些公式假定每个输出像素与其他所有输出像素条件独立，输出空间被当作无结构的空间。cGANs 能够学习结构化损失，并在输出结果的基础上生成结构损失惩罚。许多使用了条件随机场[10]、SSIM 度量[56]、特征匹配[15]、非参数化损失[37]、卷积伪先验[57]和基于匹配统计协方差损失[30]的方法将结构损失纳入考虑范围。与上述方法不同的是，cGANs 的损失函数是学习得到的，理论上能够惩罚输入和输出之间所有结构差异。

cGANs 我们不是第一个在 GANs 中加入先验条件的。有许多将 cGANs 用于离散标签[41,23,13]、文本[46]和图像的文章。图像条件模型将图像预测和普通的映射[55]、特征帧预测[40]、产品图像生成[59]和从稀疏表示生成图像[31,48]区分开。有几篇论文也将 GANs 用于图像-图像映射任务，但是没有将先验作为 GANs 的输入，而是依赖于其他项（比如 L2 回归）强迫输入与输出之间产生结构对应。这些论文在图像修补[43]、特征状态预测[64]、用于约束导向的图像操控[65]、风格迁移[38]和超分辨率[36]方面取得了相当好的结果，每个方法都是为特定任务量身定制。与之不同的是，我们的框架中所有东西都是通用的，这使我们模型的配置比其他方法简单了不少。

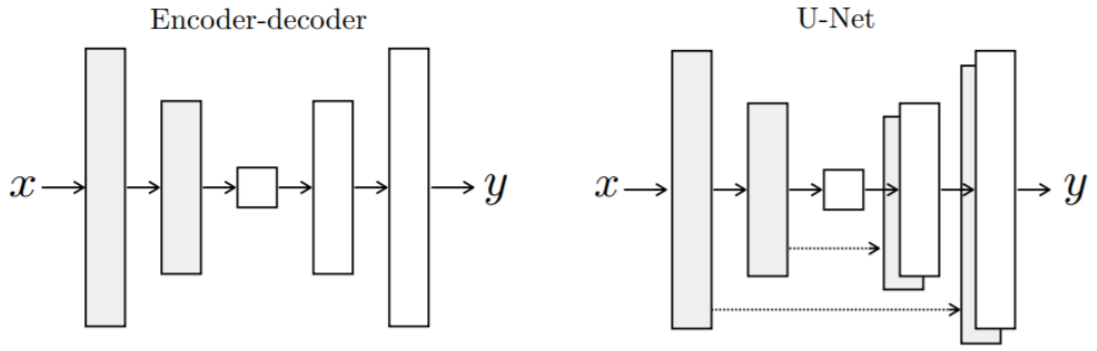


图 3 两种可供选择的生成器结构。U-Net 是在编码器-解码器基础上在编码器和解码器对应层上加入跳级连接（skip connection）

我们的方法在几个生成器和判别器的结构选择上也与其他方法不同。我们的生成器使用

基于“U-Net”的结构[50]，判别器使用卷积“PatchGAN”分类器，PatchGAN 分类器只惩罚尺度对应的图像上的结构损失。一个与 PatchGAN 相似的结构[38]曾经被用于捕获本地统计风格。这里将展示我们的方法在图像-图像转换任务中的有效性，并且我们还研究了不同 patch size 产生的不同效果。

3. 方法

GANs 是一个学习从随机噪声向量 z 映射到输出图像 y 的生成模型，用数学语言表达为：$G:\mathrm{z}\to y$。与之相对应，cGANs 学习从观察图像 x 和随机噪声向量 z 映射到输出图像 y，其数学形式：$G:\{x,z\}\to y$。生成器 G 的目的是生成判别器 D 难以区分真假的图像，而判别器 D 的目的是要尽量将生成图像和原始图像区分开。训练过程见图 2。

3.1. 目标函数

cGANs 的目标函数可表示为：

$$\mathcal{L}_{cGAN}(G,D)=E_{x,y}[\log D(x,y)]+E_{x,z}[\log(1-D(x,G(x,z))] \tag{1}$$

其中 G 被训练来最小化目标函数，D 则尽力最大化损失函数。

为了检验将先验条件作为判别器输入的效果，我们也比较了判别器无法看到观测值 x 的一个无条件变体：

$$\mathcal{L}_{GAN}(G,D)=E_{y}[\log D(y)]+E_{z}[\log(1-D(G(x,z))] \tag{2}$$

有研究发现将 GANs 损失函数和一个相对传统的损失（比如 L2 距离）结合起来对网络训练有帮助[43]。判别器的工作不变，但是生成器的任务不仅要让判别器不能判断真假，还要最小化输出原始图像间的 L2 距离。我们同样探索了这个方法，我们使用了 L1 距离而不是 L2 距离，因为 L1 距离产生的模糊更少：

$$\mathcal{L}_{L1}(G)=E_{x,y,z}[\|y-G(x,z)\|_1] \tag{3}$$

我们的最终目标函数是：

$$G^{*}=\arg\min_{G}\max_{D}\mathcal{L}_{cGAN}(G,D)+\lambda\mathcal{L}_{L1}(G) \tag{4}$$

网络在没有噪声向量 z 的时候还是能够学习从 x 到 y 的映射，但是将会生成更确定的结果，因不能匹配除 delta 函数之外的任何分布。已有的关于 cGANs 的论文[55]已经证明了这一点，除了输入 x 外还提供了输入噪声向量 z 作为生成器输入。在前期的实验中，由于生成器只是简单地忽略了噪声，我们和 Mathieu[40]的实验结果一样没有发现这个方法的有效性。相反，在我们的最终版模型中，我们只是在训练和测试的时候将噪声以 dropout 的形式应用在生成器的前几层中。尽管输入了噪声，我们在网络输出中观测到了很小的随机性。设计生成高随机性输出并能捕获条件分布中所有熵的 cGANs 值得研究的问题。

3.2 网络架构

我们网络中生成器和判别器架构从[44]中的架构改造而来。生成器和判别器都使用 convolution-BatchNorm-ReLu[29]形式的模块。补充材料提供网络架构更多细节，下面讨论网络的主要特征。

3.2.1 含跳级的生成器

图像-图像转换任务的关键特征是从高分辨率输入图像映射到高分辨率输出图像。此外，

就我们所考虑的问题而言，输入和输出虽然表现形式不同，但都是同一个底层结构的不同的表现形式，因此输出结果中的结构和输入中的结构应该大致对齐，这是我们设计生成器结构的主要依据。

在图像-图像转换任务中已经有很多工作[43,55,30,64,59]使用了编码器-解码器网络[26]。在这种网络种，输入经过一系列下采样层，直到即将反向处理的瓶颈层。输出结果中的所有信息必须流经包括瓶颈处在内的所有层。许多图像-图像转换问题希望输入层和输出层能够共享低级信息，从而使用跳级连接（skip connection）连接编码器和解码器对应层。例如在图片上色的案例中，输入和输出共享边缘的位置信息。

为了让生成器突破瓶颈处不能传播低级信息的限制限制，我们模仿 U-Net[50]的通用架构加入了跳级（skip connection）机制。确切地说，我们在每对第 i 层和第 $n-i$ 层之间加入了跳级连接（ n 是总层数）。每个跳级连接知识简单地将第 i 层和第 $n-i$ 层的所有通道串联起来。

3.2.2 **马尔可夫判别器**（PachGAN）

如图 4 所示，众所周知 L1 和 L2 距离在图像生成任务中将产生模糊[34]。尽管样的损失函数不能使输出结果中高频突出，但是在许多情况还是准确捕获低频特征。图像-图像任务不需要全新的框架来加强低频正确性，L1 距离是最好的选择。

L1 距离（等式 4）激励限制 GANs 的判别器对高频建模。为了对高频进行有效建模，我们将我们模型的注意力限制在本地图像块的结构上。因此，我们设计了一个我们叫做 PatchGAN 的只惩罚某一个区域的结构损失的判别器架构，该判别器尝试分类将每个 $N \times N$ 的图像块分类。训练时我们让图像通过了判别器的所有卷积层，取所有响应的平均值作为判别器 D 的最终输出。

在 4.4 节，我们证明模型在 N 比图像尺寸小很多的情况下还是能生成高质量的结果。这样做的优点是尺寸更小的 PatchGAN 有更少的参数、运行得更快并且可以应用于任何尺寸的图像。

判别器假设距离大于 N 的像素之间相互独立，对图像进行马尔可夫随机域建模。[38]曾经使用了这种方法，这种方法纹理建模[17,21]和风格建模[16,25,22,37]方面同样有效[17,21]。

3.3. **优化和推断**

为了优化我们的网络，我们使用了[24]中提供的的标准方法：我们选择了在判别器 D 上更新一个梯度，在生成器 G 上更新一个梯度的策略。提出 GANs 的论文[24]中说到训练生成器网络最小化 $\log(1-D(x,G(x,z)))$ 并不是最佳策略，因此我们选择训练生成器最大化 $\log D(x,G(x,z))$ 。此外，在生成判别器 D 的梯度的时候我们将损失函数除以 2 来降低判别器 D 相对于生成器 G 的学习速率。训练时我们使用了 minibatch SGD 和 Adam 优化器[32]，学习率为 0.0002，冲量常数 $\beta_1 = 0.5, \beta_2 = 0.999$ 。

在推断阶段，我们以和训练阶段完全相同的方式运行生成器。我们在测试时同样使用了 dropout，并且使用了测试样例的统计数据而不是训练时汇总统计数据来应用 batch-normalization[29]，这一点与常规方法有所不同。当 batch-size 被设为 1 的时候，batch-normalization 的方法就称为 instance-normalization，这种方法已经被证明在图像生成任务[54]中的有效性。在我们的实验中，我们根据实验在 1-10 之间选择一个数字作为 batch-size。

4. 实验

为了探索 cGANs 的泛化能力，我们使用了很多数据集进行了很多实验测试我们的模型，包括图形任务（比如照片生成）和视觉任务（比如语义分割）：

语义→图像：使用 Cityscapes 数据集[12]训练。

结构→图像：使用 CMP Facades 数据集[45]训练。

地图→航拍照片：使用从谷歌地图获取的数据训练。

BW→彩色图像：使用[51]中的数据集训练。

边缘→图像：使用来自[65]和[60]的数据集训练，二分边缘使用 HED 边缘检测器[58]生成并加以后处理。

素描→图像：用于测试由边缘生成图像模型，来自于[19]的人工画的素描图。

白天→夜晚：使用来自于[33]数据集训练。

热敏图→彩色图：使用来自于[27]的数据集训练。

像素缺失图像→补全的图像：使用来自[14]的巴黎街景训练。

网络补充材料提供了使用上述数据集训练的细节。在所有情况中，输入和输出都是 1-3 通道的图像。定性结果在图 8、9、10、11、12、13、14、15、16、17、18、19、20 中展示。几个失败的例子在图 21 中。更多材料见 https://github.com/phillipi/pix2pix。

数据需求和速度 尽管很多任务中只使用了非常小的数据集进行训练，我们还是能够获得非常好的结果。我们正面训练数据集只含 400 个图像（结果见图 14），白天到黑夜转换的训练集只含有 91 个网络图像（结果见图 15）。在这些较小的数据集上，训练过程可以非常快，例如图 14 中展示的结果在一张 Pascal Titan X GPU 上训练只花了不到两小时。测试时，所有模型在这个 GPU 上运行非常好。

Loss	Per-pixel acc.	Per-class acc.	Class IOU
L1	0.42	0.15	0.11
GAN	0.22	0.05	0.01
cGAN	0.57	0.22	0.16
L1+GAN	0.64	0.20	0.15
L1+cGAN	**0.66**	**0.23**	**0.17**
Ground truth	0.80	0.26	0.21

表 1　不同损失函数使用 Cityscapes 数据集评估便标签与图像相互转换的 FCN 指标

Loss	Per-pixel acc.	Per-class acc.	Class IOU
Encoder-decoder (L1)	0.35	0.12	0.08
Encoder-decoder (L1+cGAN)	0.29	0.09	0.05
U-net (L1)	0.48	0.18	0.13
U-net (L1+cGAN)	**0.55**	**0.20**	**0.14**

表 2 不同生成器结构和目标函数的 FCN 指标，使用在 Cityscapes 数据集上进行标签与图像相互转换评估（U-Net(L1-cGAN)与其他表中报告不同因为本实验中 batch size 是 10，其他实验是 1）

Discriminator receptive field	Per-pixel acc.	Per-class acc.	Class IOU
1×1	0.39	0.15	0.10
16×16	0.65	0.21	**0.17**
70×70	**0.66**	**0.23**	**0.17**
286×286	0.42	0.16	0.11

表 3 判别器不同接受域大小的 FCN 指标，使用 Cityscapes 数据集标签转图像评估。输入图像分辨率是 256×256 。

4.1. 指标评估

评估生成图像的质量是一个困难的问题[52]。传统的指标例如像素均方误差并不评估结果的联合统计数据，因此并不能度量结果中的结构化损失。

	Photo → Map	Map → Photo
Loss	% Turkers labeled *real*	% Turkers labeled *real*
L1	2.8% ± 1.0%	0.8% ± 0.3%
L1+cGAN	6.1% ± 1.3%	**18.9% ± 2.5%**

表 4 使用地图与航拍图相互转换结果进行 AMT 真假测试

为了更全面地评估我们结果的视觉质量，我们使用了两种策略。首先，我们使用 Amazon Mechanical Turk（AMT）运行了“真和假”的感知研究。对于类似图像上色和图像生成这样的图像问题，人眼观察的合理性往往是最终目标。因此，我们使用这种方法测试我们的地图生成、航拍图像生成和图像上色任务。

我们通过现有的识别系统是否能识别图像中的物体来度量我们生成的 cityscapes 图像是否足够真实。这个指标与[52]中的 inception 指标、[55]中的物体检测评估、[62]和[64]中的语义可解释性指标类似。

Method	% Turkers labeled *real*
L2 regression from [62]	16.3% ± 2.4%
Zhang et al. 2016 [62]	**27.8% ± 2.7%**
Ours	22.5% ± 1.6%

表 5 图像上色任务中 AMT 真假测试

AMT 感知研究 我们的 AMT 实验按照[62]中的协议进行，使用 Turkers 进行了一系列真假图像对比的实验。在每次测试中，每张图像出现 1 秒钟时间，之后 Turkers 可以在不受限制的时间内对该图像的真假进行判断。每个会话的前 10 张图像用于测试，每张图像都带有相应的标签。实验的主要部分包含 40 个样例，每个图像都没有给定标签。每个会话每次只测试模型在一个任务上的表现，并且不允许 Turkers 同时进行多个会话。每个任务大约使

用 50 个 Turkers 评估。对于图像上色实验，真假图像都来自于输入灰度图像。对于地图转换为航拍图任务，为了让任务更难且避免生成像地上一样的结果，真假图像不是由同一个输入图像生成。对于地图转换航拍图的任务，我们使用分辨率为 256×256 的图像进行训练，但是利用全卷积转换可以使用 512×512 的图像进行测试。该图像在使用 Turkers 测试前进行下采样生成分辨率为 256×256 的图像。对于图像上色任务，我们使用分辨率为 256×256 的图像进行训练和测试，并且在 Turkers 中使用了相同的分辨率表示结果。

全卷积评估 尽管定量地评估生成模型非常困难，最近的论文[52,55,62,42]尝试了使用预训练的语义分类器来评估生成结果的可判别性作为伪度量。如果合成图像中含有输入中大部分结构，使用真实图像训练的分类器将能够正确地将合成图像分类。为此，我们采用了使用比较多的 FCN-8s[39]结构用于语义分割，并且使用 cityscapes 数据集进行训练。然后通过对合成图像的分类准确率和原图像分类准确率对合成图像进行评估。

4.2. 目标函数分析

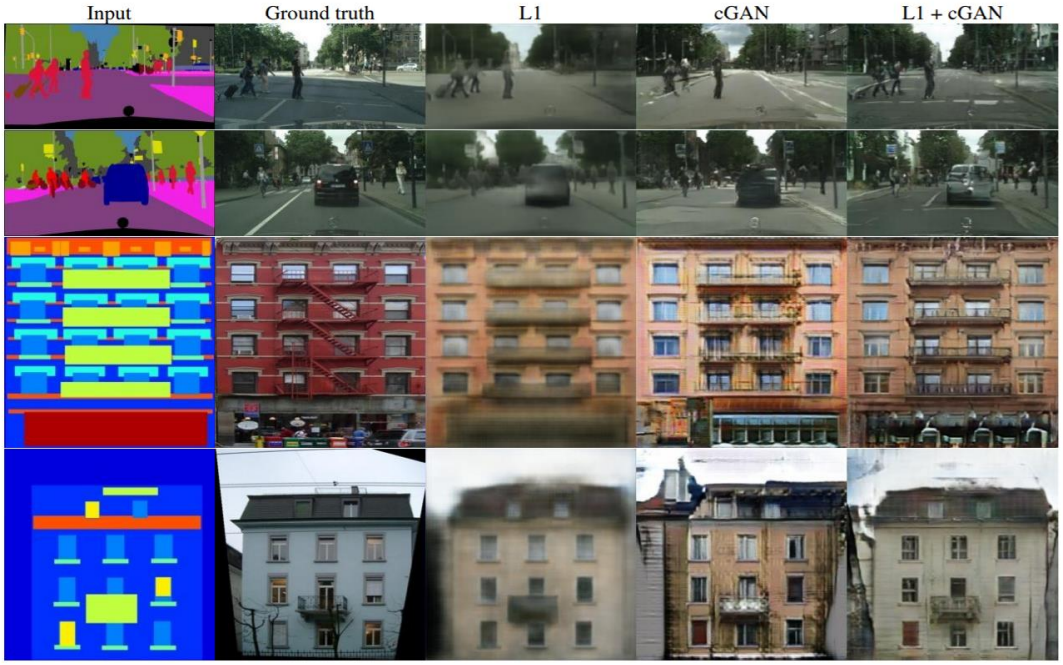


图 4 不同的损失函数得到不同质量的结果。每一列展示使用一个损失函数训练的结果。

公式 4 中那种成分更重要呢？我们做了剥离实验来研究 L1 正则项（L1 term）、GANs 项独立的效果，并且比较了条件判别器（cGANs, 公式 1）和非条件判别器（GANs, 公式 2）的实验结果。

图 4 展示了这几种变体在完成标签-图像任务的定量效果。单独使用 L_1 正则项得到了合理的结果，但是生成的图像很模糊。单独使用 cGAN（设置公式 4 中 $\lambda=0$）能得到更合理的结果但是在某些应用中结果含有肉眼可见的合成痕迹。将两项加在一起（$\lambda=100$）可以减少生成结果中的合成痕迹。

我们使用 FCN 指标量化使用 cityscape 数据集进行标签-图像转换任务的结果（见表 1）：基于 GAN 的目标函数得分比其他目标函数高，这表明合成图像包含更过可识别结构。我们也进行了从判别器输入中移除先验条件项的实验。在这种情况下，损失函数对于输入和输出

之间的不匹配没有惩罚，而是只关心生成图像的真实性。这个变体表现非常差，检查结果发现不论输入什么图像，生成器都生成了相同的输出。cGAN 的表现明显由于 GANs，说明损失函数衡量输入和输出的匹配度非常重要。然而，添加 L_1 项也能增加输入和输出的关联性，因为 L_1 项会对输入和输出的距离进行惩罚，只有正确地匹配并且合成了输出才能不受到惩罚。相应地，L1+GANs 组合在标签-图像任务中同样有效。

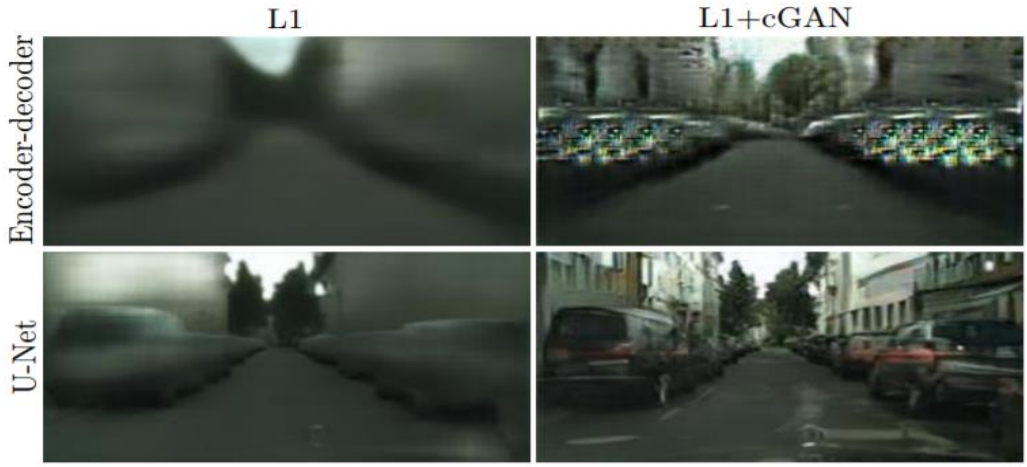
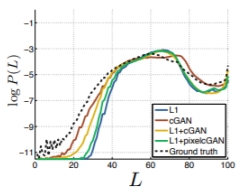
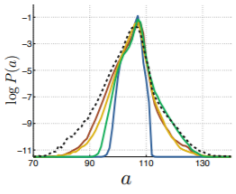


图 5　在编码器-解码器结构中加入跳级连接（skip connection）生成更高质量的结果。

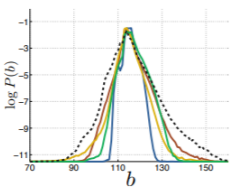
色彩丰富度 条件 GANs 的一个令人震惊的效果是它可以生成清晰的图像，并且在输入标签图空白的地方甚至会进行适当的想象，可以想象的是 cGANs 让图像在光谱维度上变得清晰中有相似的效果，例如让图像颜色更加丰富。而仅使用 L1 距离作为损失函数的生成器在不知道将边缘放在什么地方的时候将生成模糊的图像，同样当生成器不确定某个像素使用哪种颜色值比较合理的时候也会使用灰色作为像素值，原因是使用所有颜色的条件概率密度的平均值将会使 L1 距离最小化。相原则上加入一个对抗损失将会使模型知道灰色的输出是不真实的，从而让输出和真实的颜色分布更匹配[24]。在图 6 中，我们检验了在 Cityscapes 数据集上我们的 cGANs 模型是否能够达到理想的效果。图像显示在 Lab 颜色空间的输出颜色分布（真实分布使用虚线表示）。很明显 L1 将会使模型输出一个比真实值窄的颜色分布，证明了 L1 将会导致生成平均、灰色图像。相反，使用 cGAN 让输出分布于真实分布更近。



(a)



(b)



(c)

Loss	Histogram intersection against ground truth L	a	b
L1	0.81	0.69	0.70
cGAN	**0.87**	0.74	**0.84**
L1+cGAN	0.86	**0.84**	0.82
PixelGAN	0.83	0.68	0.78

(d)

图 6　cGAN 的颜色分布匹配属性

4.3 生成器结构分析

U-Net 结构允许低级的信息跨过网络直接传递，这样会使结果更优嘛？图 5 和表 2 比较了 U-Net 结构和编码器-解码器结构在 cityscape 数据集生成上结果。编码器-解码器结构与

U-Net 结构比起来只减少了跳级连接（skip connection）。在我们的实验中编码器-解码器结构不能学习到生成真实图像。U-Net 的优点不限于使用 cGANs：当只使用 L1 距离作为损失函数训练 U-Net 和编码器-解码器时，U-Net 输出的结果同样更为理想。

4.4 PixelGANs PatchGANs ImageGANs

我们测试不同判别器接收域 N 的效果，从 1×1 的“PatchGANs”到 286×286 的“ImageGANs”，图 6 展示图像生成效果，表 3 是不同 N 大小的模型 FCN 指标评估结果。在本文中除了特殊表明外，我们使用的都是大小为 70×70 的 PatchGANs，本节中所有实验使用 L1+cGANs 损失函数。

PixelGANs 的接收域 N 的大小对空间结构的清晰度没有影响，但是可以增加结果的颜色丰富程度（结果见图 6）。例如使用 L1 距离作为损失的时候图 7 中的公交车被上色为灰色，但是使用 PixelGANs 损失函数的时候却被上色为红色。颜色直方匹配是图像处理中的共同问题[49]，PixelGANs 有望成为轻量级的解决方法。

图 7　Patch size 变量。不确定区域在使用 L1 损失函数的时候变得模糊。1×1 的 PixelGAN 生成更大的颜色多样性，但是对空间统计没有影响。16×16 的 PatchGAN 得到局部清晰的结果。70×70 的 PatchGAN 生成的结果比较清晰的，尽管空间和色彩空间不正确。286×286 的 ImageGAN 生成与 70×70 的 PatchGAN 相似的结果，但是使用 FCN 指标衡量时其表现稍微差一些。

使用 16×16 的 PatchGANs 足够得到比较清晰的图像和得到比较高的 FCN 分数，但是也有少许人造的痕迹，70×70 的 PatchGANs 减轻这些人造痕迹，并且得到了更好的 FCN 分数。此外，全尺寸的 286×286 的 ImageGANsb 不能明显提高结果的视觉质量，实际上使用 ImageGANs 的模型的 FCN 分数还有所下滑（见表 3）。这可能是因为 ImageGANs 比 70×70 的 PatchGANs 有更多参数所以更难训练。

图 8　使用谷歌地图 512×512 像素的案例（训练模型是使用的是分辨率为 256×256 的图像）

全卷积转换 PatchGANs 的一个优点是固定输入尺寸的判别器可以被用于任意大的图像上。我们也可以将不大于生成器输入尺寸的图像应用到生成器中。我们使用地图-航拍图转换任务不同尺寸的输入效果。在使用 256×256 的图像训练生成器后使用大小为 512×512 的图像测试。图 8 中的结果证明了这个方法的有效性。

4.5 感知验证

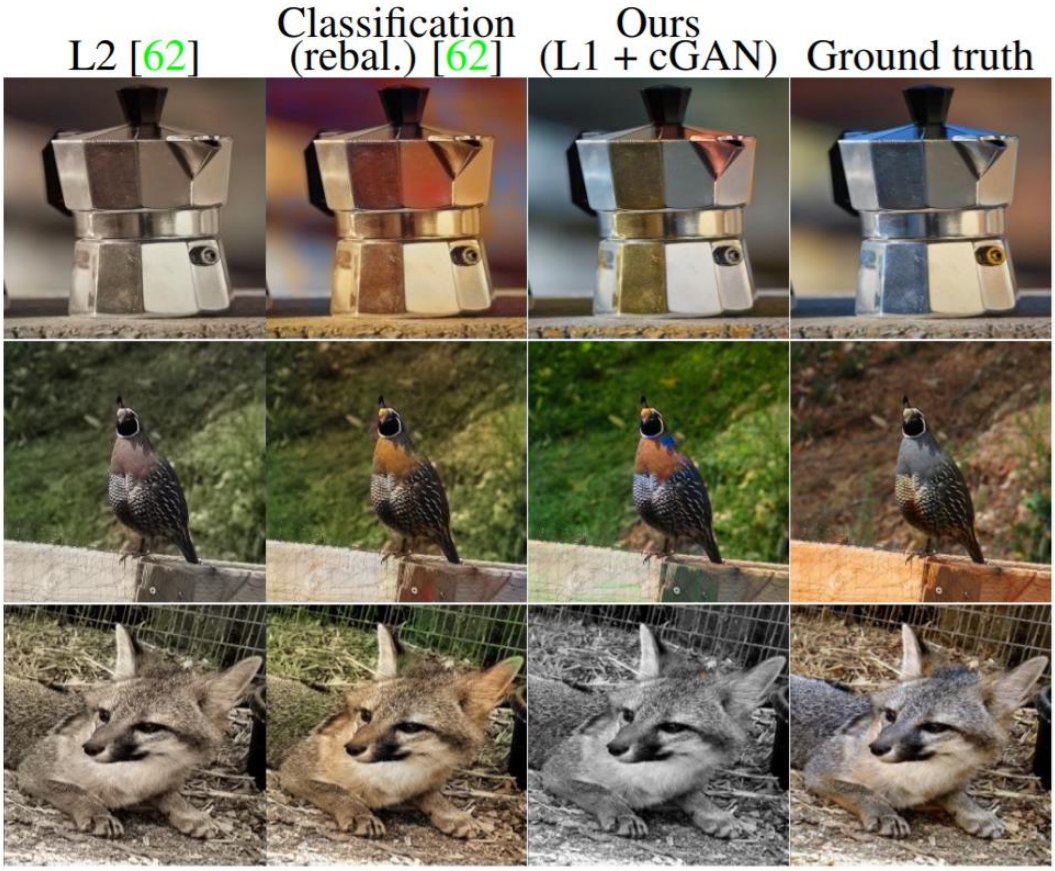


图 9 cGANs 和来自[62]的 L2 回归版以及[64]的完整版。cGANs 生成引人注目的上色结果。

我们使用地图-航拍图转换任务和灰度图-彩色图转换任务进行生成图像的感知测试。表 4 给出了在 AMT 上做的地图-航拍图转换任务的实验结果。模型生成的航拍图欺骗了 18.9%的实验参与者，而 L1 基线由于生成模糊图像几乎不能欺骗任何实验参与者。而模型使用航拍图生成的地图上只欺骗了 6.1%实验参与者，并没有明显由于 L1 基线。这可能是因为在地图上很小的结构错误很容易发现，因为地图比航拍图拥有更规则的几何形状。

我们使用 ImageNet[51]训练模型进行图像上色并且使用[62,35]中提供的测试子集来测试。我们使用 L1+cGANs 损失的方法欺骗了 22.5%的参与者（表 5）。我们同样也测试了[62]的结果和他们使用 L2 损失的变体。cGANs 与[62]中 L2 损失的变体表现不相上下，但是落后于[62]的完整版。我们认为他们的方法已经具体化用于图像上色任务了所以表现比较好。

4.6 语义分割

cGANs 在输出或图像上需要更多细节的任务上表现更好，在图像处理或图形任务上也

是一样。那么像语义分割这样的输出比输入更简单的视觉问题表现如何呢？

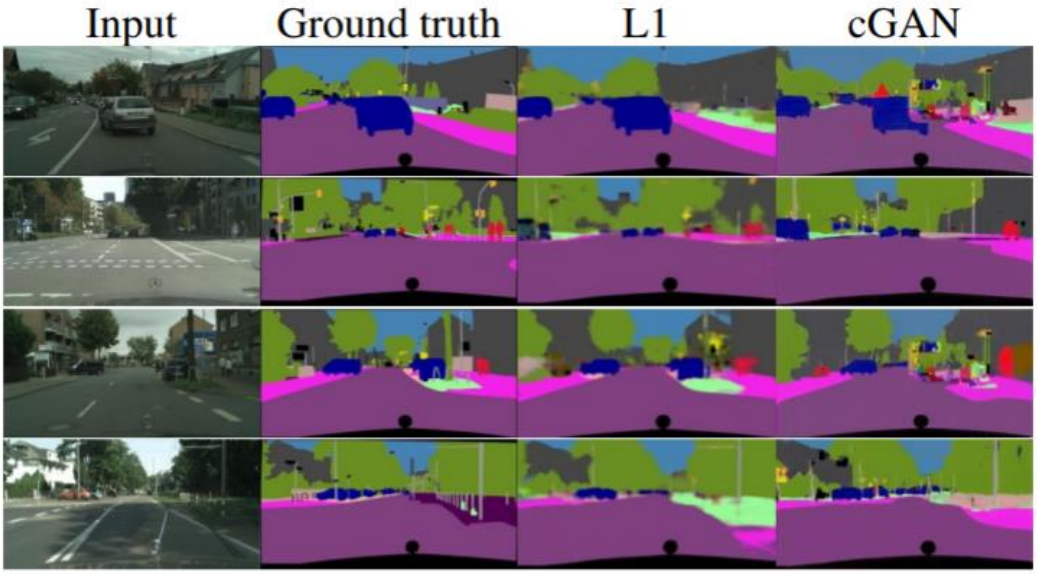


图 10 将 cGANs 应用于语义分割。cGAN 生成相对较清晰的看起来像真实标注图像的图像，但实际上包含许多小的想象的物体。

为了测试这个指标，我们在 cityscape 数据集上训练了一个图像-标签的 cGANs（包含和不包含 L1 损失）模型。图 10 展示了模型结果，相应的量化指标（准确率）见表 6。有趣的是，不使用 L1 距离作为损失的 cGAN 模型能够以较高的准确率解决这个问题。就我们所知，这是首次证明 GANs 成功生成了几乎离散的标签，与图像的像素值连续变化不同。尽管 cGAN 取得了比较好的成就，但是它比现有最好的方法差的太远：简单地使用 L1 回归得到的结果比 cGAN 更好（见表 6）。对于视觉任务，目标（预测输出接近真实值）可能比图形任务更清晰，大部分情况下像 L1 距离这样的重建损失就足以应对。

Loss	Per-pixel acc.	Per-class acc.	Class IOU
L1	**0.86**	**0.42**	**0.35**
cGAN	0.74	0.28	0.22
L1+cGAN	0.83	0.36	0.29

表 6 在 cityscape 数据集上进行图像-标签任务的表现

4.7 社区驱动研究

自从我们的论文和 pix2pix 代码库的发布，包括计算机视觉、计算机图形从业者和视觉艺术家在内的推特社区已经成功地将我们的框架应用于比我们原文中的领域多得多的图像到图像转换任务。图 11 展示了几个应用 pix2pix 的例子，包括背景移除、调色板生成、素描转肖像、素描转宠物、姿势转换等。由于这些创新项目不是在可控和科学条件下的工作，可能需要对 pix2pix 代码的修改。尽管如此，这些项目证明了我们的方法有作为图像-图像转换任务的通用工具的可能。

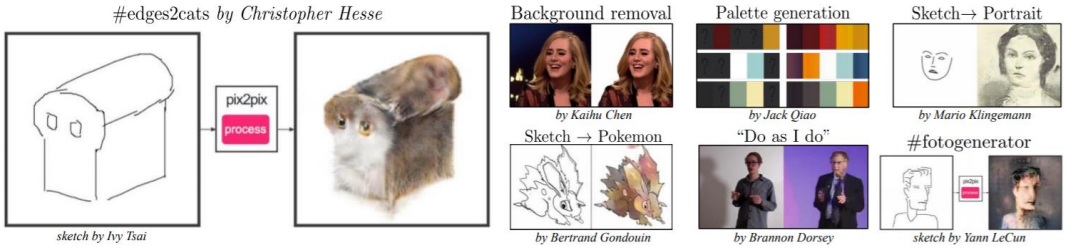


图 11 将 cGANs 应用于其他任务的例子

5．结论

这篇论文的结果证明条件对抗网络有希望成为许多图像-图像转换任务的通用工具，特别是那些需要高度结构化图形输出的任务。条件对抗网络可以学习一个应用于当前任务和数据的损失函数，这让模型在很多任务中有用武之地。

参考文献

[1] Bertrand gondouin. https://twitter.com/ bgondouin/status/818571935529377792. Accessed, 2017-04-21. 9

[2] Brannon dorsey. https://twitter.com/ brannondorsey/status/806283494041223168. Accessed, 2017-04-21. 9

[3] Christopher hesse. https://affinelayer.com/ pixsrv/. Accessed: 2017-04-21. 9

[4] Gerda bosman, tom kenter, rolf jagerman, and daan gosman. https://dekennisvannu.nl/site/artikel/ Help-ons-kunstmatige-intelligentie-testen/ 9163. Accessed: 2017-08-31. 9

[5] Jack qiao. http://colormind.io/blog/. Accessed: 2017-04-21. 9

[6] Kaihu chen. http://www.terraai.org/ imageops/index.html. Accessed, 2017-04-21. 9

[7] Mario klingemann. https://twitter.com/ quasimondo/status/826065030944870400. Accessed, 2017-04-21. 9

[8] Memo akten. https://vimeo.com/260612034. Accessed, 2018-11-07. 9

[9] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In CVPR, 2005. 1

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In ICLR, 2015. 2

[11] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: internet image montage. ACM Transactions on Graphics (TOG), 28(5):124, 2009. 1

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In CVPR, 2016. 4, 16

[13] E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In NIPS, 2015. 2

[14] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros. What makes paris look like paris? ACM Transactions on Graphics, 31(4), 2012. 4, 13, 17

[15] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In NIPS, 2016. 2

[16] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In SIGGRAPH, 2001. 1, 4

[17] A. A. Efros and T. K. Leung. Texture synthesis by nonparametric sampling. In ICCV, 1999. 4

[18] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In ICCV, 2015. 1

[19] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? In SIGGRAPH, 2012. 4, 12

[20] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. ACM Transactions on Graphics (TOG), 25(3):787– 794, 2006. 1

[21] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks.

In NIPS, 2015. 4

[22] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. CVPR, 2016. 4

[23] J. Gauthier. Conditional generative adversarial nets for convolutional face generation. Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester, (5):2, 2014. 2

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014. 2, 4, 6, 7

[25] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In SIGGRAPH, 2001. 1, 4

[26] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 313(5786):504–507, 2006. 3

[27] S. Hwang, J. Park, N. Kim, Y. Choi, and I. So Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In CVPR, 2015. 4, 13, 16

[28] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. ACM Transactions on Graphics (TOG), 35(4), 2016. 2

[29] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015. 3, 4

[30] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In ECCV, 2016. 2, 3

[31] L. Karacan, Z. Akata, A. Erdem, and E. Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. arXiv preprint arXiv:1612.00215, 2016. 2

[32] D. Kingma and J. Ba. Adam: A method for stochastic optimization. ICLR, 2015. 4

[33] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. ACM Transactions on Graphics (TOG), 33(4):149, 2014. 1, 4, 16

[34] A. B. L. Larsen, S. K. Sønderby, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In ICML, 2016. 3

[35] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. ECCV, 2016. 2, 8, 16

[36] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In CVPR, 2017. 2

[37] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. CVPR, 2016. 2, 4

[38] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. ECCV, 2016. 2, 4

[39] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015. 1, 2, 5

[40] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. ICLR, 2016. 2, 3

[41] M. Mirza and S. Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014. 2

[42] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. In CVPR, 2016. 5

[43] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In CVPR, 2016. 2, 3, 13, 17

[44] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR, 2016. 2, 3, 16

[45] R. S. Radim Tyle ˇ cek. Spatial pattern templates for recogni- ˇ tion of objects with regular structure. In German Conference on Pattern Recognition, 2013. 4, 16

[46] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In ICML, 2016. 2

[47] S. Reed, A. van den Oord, N. Kalchbrenner, V. Bapst, M. Botvinick, and N. de Freitas. Generating interpretable images with controllable structure. In ICLR Workshop, 2017. 2

[48] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In NIPS, 2016. 2

[49] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. IEEE Computer Graphics and Applications, 21:34–41, 2001. 7

[50] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In MICCAI, 2015. 2, 3

[51] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3):211–252, 2015. 4, 8, 16

[52] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In NIPS, 2016. 2, 4, 5

[53] Y. Shih, S. Paris, F. Durand, and W. T. Freeman. Data-driven hallucination of different times of day from a single outdoor photo. ACM Transactions on Graphics (TOG), 32(6):200, 2013. 1

[54] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022, 2016. 4

[55] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In ECCV, 2016. 2, 3, 5

[56] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, 2004.

2

[57] S. Xie, X. Huang, and Z. Tu. Top-down learning for structured labeling with convolutional pseudoprior. In ECCV, 2015. 2

[58] S. Xie and Z. Tu. Holistically-nested edge detection. In ICCV, 2015. 1, 2, 4

[59] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon. Pixellevel domain transfer. ECCV, 2016. 2, 3

[60] A. Yu and K. Grauman. Fine-Grained Visual Comparisons with Local Learning. In CVPR, 2014. 4

[61] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In CVPR, 2014. 16

[62] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. ECCV, 2016. 1, 2, 5, 7, 8, 16

[63] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. In ICLR, 2017. 2

[64] Y. Zhou and T. L. Berg. Learning temporal transformations from time-lapse videos. In ECCV, 2016. 2, 3, 8

[65] J.-Y. Zhu, P. Krahenb ¨ uhl, E. Shechtman, and A. A. Efros. ¨ Generative visual manipulation on the natural image manifold. In ECCV, 2016. 2, 4, 16