

少样本无监督的图像到图像转换 (FUNIT)

Ming-Yu Liu¹, Xun Huang^{1,2}, Arun Mallya¹, Tero Karras¹, Timo Aila¹, Jaakko Lehtinen^{1,3}, Jan Kautz¹

¹NVIDIA, ²Cornell University, ³Aalto University

fmingyul, xunh, amallya, tkarras, taila, jlehtinen, jkautz@google.com

摘要

无监督的图像到图像转换方法学习将给定类中的图像映射到不同类中的类似图像，利用图像的非结构化（无标记）数据集。虽然非常成功，但是当前的方法需要在训练时访问源类和目标类中的许多图像。我们认为这极大地限制了它们的使用。从人类具备的能力从少数几个例子中发现新物体本质的能力中汲取灵感，并从这一点推广，我们寻求一种少样本无监督的图像到图像的转换算法，在测试时指定的目标类仅由少量示例图像组成。我们的模型通过将对抗性训练方案与新颖的网络设计相结合来实现这种只经少数迭代的生成能力。通过广泛的实验验证和与基准数据集的几种基线方法的比较，我们验证了所提出的框架的有效性。代码在下面地址获取：

<https://nvlabs.github.io/FUNIT>.

1. 介绍

人类在泛化方面非常擅长。当给出一张以前看不见的外来动物的图片时，我们可以在不同的姿势下形成同一动物的生动的心理描绘图像，特别是当我们遇到（之前）相似但不同的动物之前。例如，一个人第一次看到一只站立的老虎，可以毫不费力地想象它会看起来躺着的样子，因为它有着其他动物的一些经验作对照。

虽然最近的无监督图像到图像转换算法在跨图像类[30,45,29,25,54,51]传递复杂的外观变化方面非常成功，但是能够从基于新类的少数样本中推广出来是很受限的，因为先验知识完全超出他们的能力范围。具体地说，他们需要对他们要进行转换的所有类别的图像进行大量训练，即，它们不支持少样本生成。

为了弥合人类和机器想象能力之间的差距，我们提出了少样本

无监督的图像到图像转换 (FUNIT) 框架工作，旨在学习图像到图像的转换模型，通过利用目标的少量图像将源类图像映射到目标类的分析图像在测试时给出的类别模式。该模型在训练期间从未显示目标类的图像，但要求在测试时生成其中一些。为了继续，我们首先假设人类的少数生成能力是从他们过去的视觉经验中发展出来的 - 如果一个人在过去看过更多不同的对象类，他们可以更好地想象一个新对象的视图。基于该假设，我们使用包含许多不同对象类的图像的数据集来训练我们的 FUNIT 模型，以模拟过去的视觉体验。具体来说，我们通过利用另一个类的少量示例图像来训练模型以将图像从一个类转换为另一个类。我们假设通过学习从用于转换任务的少数示例图像中提取外观模式，该模型学习了一种可推广的外观模式提取器，其可以在测试时针对少样本图像应用于看不见的类的图像 - 图像转换任务。在实验部分，我们给出了直觉的证据，即随着训练集中类的数量增加，少样本转换性能得到改善。

我们的框架基于 Generative Adversarial Networks (GAN) [14]。我们表明，通过将对抗训练方案与新颖的网络设计相结合，我们实现了所需的少样本无监督的图像到图像的转换能力。通过对三个数据集的广泛实验验证，包括使用各种性能指标与几种基线方法的比较，我们验证了我们提出的框架的有效性。另外，我们展示了所提出的框架可以应用于少样本图像分类任务。通过训练我们的模型为少样本类别生成的图像上的分类器，我们能够胜过基于特征幻觉 (feature hallucination) 的最先进的少样本分类方法。

2. 相关工作

无监督/不成对的图像到图像转换旨在学习条件图像生成函数

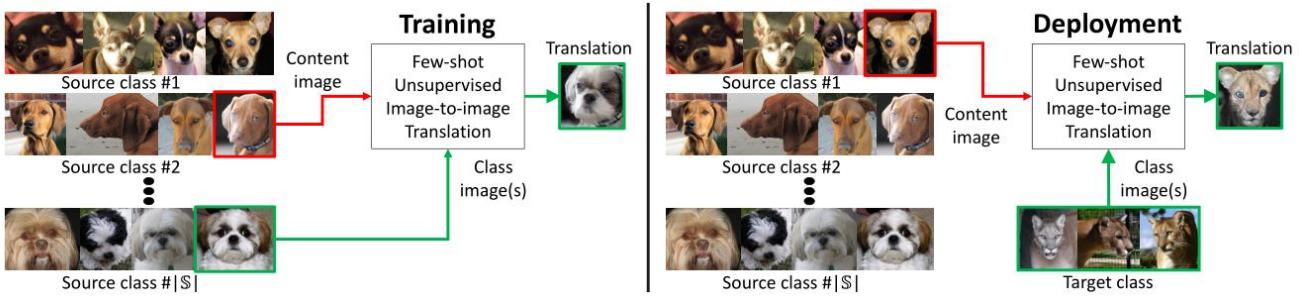


图 1.训练。 训练集由各种对象类（源类）的图像组成。我们训练模型来在这些源对象类之间转换图像。**部署。** 我们向训练模型显示目标类的极少数图像，这足以将源类的图像转换为目标类的类似图像，即使模型在训练期间从未见过来自目标类的单个图像。请注意，FUNIT 生成器需要两个输入：1) 内容图像和 2) 一组目标类图像。它旨在生成类似于目标类图像与输入图像间的转换。

能够将源类的输入图像映射到目标类的模拟图像而无需成对监督。这个问题本质上是不适合的，因为它试图使用来自边际分布的样本重新覆盖联合分布[29,30]。为了解决这个问题，现有的工作使用了额外的约束。例如，一些工作强制执行转换以保留源数据的某些属性，例如像素值[40]，像素梯度[5]，语义特征[45]，类标签[5]或成对样本距离[3]。有一些工作强制执行循环一致性约束[51,54,25,1,55]。一些作品使用共享/部分共享的潜在空间假设[29,30] / [19,26]。我们的工作基于部分共享的潜在空间假设，但是专为少样本无监督的图像到图像转换任务而设计。

虽然能够生成逼真的转换输出，但是现有的无监督图像到图像转换模型在两个方面受到限制。首先，如果在训练时只给出很少的图像，它们的样本效率低，产生差的转换输出。其次，学习的模型仅限于在两个类之间转换图像。尽管新任务与原始任务之间存在相似性，但是用于一个转换任务的训练模型不能直接重用于新任务。例如，即使猫与老虎有很大的相似性，也不能将哈士奇与猫的转换模型重新用于哈士奇与老虎的转换。

最近，Benaim 和 Wolf [4]提出了一种无监督的图像到图像转换框架，用于部分解决第一个方面。具体而言，他们使用由一个源类图像组成的训练数据集，但是使用许多目标类图像来训练用于将单个源类图像转换为目标类的类似图像的模型。我们的工作在几个主要方面与他们的工作不同。首先，我们假设许多源类图像，但很少有目标类图像。此外，我们假设少数目标类图像仅在测试时可用，并且可以来自许多不同的对象类。

多类别无监督图像到图像转换[8,2,20]。将无监督的图像到图像转换方法扩展到多个类。基于我们的训练数据集是由多个类的图像组成这一点来说，我们的工作类似于这些方法。但是，我们不是在所看到的类中转换图像，而是专注于将这些看到的类的图像转换为以前看不见的类的类似图像。

少量类别。与少样本的图像到图像的转换不同，使用少量例子学习新类别的分类器的任务是一个长期研究的问题。早期的作品使用外观的生成模型，以分层的方式在各个阶段共享先验[11,38]。最近的工作重点是使用元学习来快速地让模型适应新的任务[12,34,37,33]。这些方法可以学习更好的训练优化策略，因此只需看几个例子即可提高性能。另一些工作侧重于学习图像嵌入，这些图像嵌入更适合于少样本学习[48,42,43]。最近的几项研究建议通过生成与新类[10,15,50]相对应的新特征向量来增加少数分类任务的训练集。我们的工作是针对少样本无监督的图像到图像的转换而设计的。但是，它可以应用于少样本分类，如实验部分所示。

3.少样本无监督图像转换

所提出的 FUNIT 框架旨在通过利用在测试时可用的一些目标类图像，将源类的图像映射到不层见过的目标类的类似图像。为了训练 FUNIT，我们使用来自一组对象类（例如各种动物物种的图像）的图像，称为源类。我们不假设任何两个类别之间配对图像的存在（即，没有两个不同物种的动物处于完全相同的姿势）。我们使用源类图像来训练多级无监督

图像到图像的转换模型。在测试过程中，我们为模型提供了一些来自新对象类的图像，称为目标类。该模型必须利用少数目标图像将任何源类图像转换为目标类的同类图像。当我们从不同的新对象类提供相同模型的少量图像时，它必须将任何源类图像转换为不同新对象类的类似图像。

我们的框架由条件图像生成器 G 和多任务对抗判别器 D 组成。与现有的无监督图像到图像转换框架[54,29]中的条件图像生成器不同，它采用一个图像作为输入，我们的生成器 G 同时采用内容图像 x 和一组类别图像 $K: \{y_1, \dots, y_K\}$ 作为输入并产生输出图像 \bar{x} 通过：

$$\bar{x} = G(x, \{y_1, \dots, y_K\}). \quad (1)$$

我们假设内容图像属于对象类 c_x ，而每个 K 类图像属于对象类 c_y 。通常， K 是一个小数字， c_x 与 c_y 不同。我们将 G 称为少样本图像转换器。

如图 1 所示， G 将输入内容图像 x 映射到输出图像 \bar{x} ，使得 x 看起来像属于对象类 c_y 的图像，并且 x 和 \bar{x} 共享结构相似性。设 S 和 T 分别表示源类集和目标类集。在训练期间， G 学习在两个随机采样的源类 c_x 之间转换图像； $c_x, c_y \in S$ ，其中 $c_x \neq c_y$ 。在测试时， G 从未看到的目标类 $c \in T$ 获取一些图像作为类图像，并将从任何源类采样的图像映射到目标类 c 的类似图像。

接下来，我们讨论网络设计和学习。更多细节见附录。

3.1. 少样本图像转换器

少样本图像转换器 G 由内容编码器 E_x ，类别编码器 E_y 和解码器 F_x 组成。内容编码器由几个 2D 卷积层组成，后跟几个残余块[16,22]。它将输入内容图像 x 映射到内容潜码 z_x ，其代表空间特征映射。类别编码器由几个 2D 卷积层组成，后面是沿样本轴的平均操作。具体地说，它首先映射 K 个类别图像 $\{y_1, \dots, y_K\}$ 中的每一个转换为中间潜在向量，然后计算中间潜在向量的平均值，以获得最终的潜码 z_y 。

解码器由几个自适应实例正规化 (AdaIN) 残余块[19]组成，后面跟着一些上采样卷积层。AdaIN 残余块是使用 AdaIN [18] 作为正则化层的残余块。对于每个样本，AdaIN 首先将每个通道中样本的激活函数标准化为零均值和单位方差。然后它会缩放激活使用

一组标量和偏置组成的学习仿射变换。注意，仿射变换在空间上是不变的，因此只能用于获得全局外观信息。通过两层全连接网络使用 z_y 自适应地计算仿射变换参数。使用 E_x ， E_y 和 F_x (此处记为 1) 被分解为

$$\bar{x} = F_x(z_x, z_y) = F_x(E_x(x), E_y(\{y_1, \dots, y_K\})). \quad (2)$$

通过使用这种转换器设计，我们的目标是使用内容编码器提取具有类不变的潜在表示（例如，对象姿势）并使用类编码器提取类特定的潜在表示（例如，对象外观）。通过经由 AdaIN 层将类别潜码馈送到解码器，我们让类图像控制全局外观（例如，对象外观），而内容图像确定局部结构（例如，眼睛的位置）。

在训练时，类编码器学习从源类的图像中提取特定类的潜在表示。在测试时，这概括为以前看不见的类的图像。在实验部分中，我们展示了泛化能力取决于训练期间看到的源对象类的数量。当 G 训练有更多的源类（例如，更多种类的动物）时，它具有更好的少样本镜像转换性能（例如，更好地将哈士奇转换成山狮）。

3.2. 多任务对抗判别器

我们的判别器 D 通过同时解决多个对抗分类任务来训练。每个任务是二分类任务，确定输入图像是源类的实际图像还是来自 G 的转换输出。由于存在 $|S|$ 源类， D 产生 $|S|$ 输出。当为源类 c_x 的实际图像更新 D 时，如果其 c_x th 输出为假，则对 D 进行处罚。对于产生源类 c_x 的伪图像的转换输出，如果其 c_x th 输出为正，则我们惩罚 D 。我们不会因为没有为其他类的图像 ($S \setminus \{c_x\}$) 预测错误而使用 D 。更新 G 时，如果 D 的 c_x th 输出为假，我们只会惩罚 G 。我们凭经验发现这种判别器比通过解决更难的 $|S|$ 级分类问题训练的判别器更有效。

3.3. 学习

我们通过解决由下式给出的极小极大优化问题来训练所提出的 FUNIT 框架：

$$\min_D \max_G \mathcal{L}_{GAN}(D, G) + \lambda_R \mathcal{L}_R(G) + \lambda_F \mathcal{L}_F(G) \quad (3)$$

其中 \mathcal{L}_{GAN} ， \mathcal{L}_R 和 \mathcal{L}_F 分别是 GAN 损失，内容图像重建损失和特征匹配损失。该

	Setting	Top1-all ↑	Top5-all ↑	Top1-test ↑	Top5-test ↑	DIPD ↓	IS-all ↑	IS-test ↑	mFID ↓
Animal Faces	CycleGAN-Unfair-20	28.97	47.88	38.32	71.82	1.615	10.48	7.43	197.13
	UNIT-Unfair-20	22.78	43.55	35.73	70.89	1.504	12.14	6.86	197.13
	MUNIT-Unfair-20	38.61	62.94	53.90	84.00	1.700	10.20	7.59	158.93
	StarGAN-Unfair-1	2.56	10.50	9.07	32.55	1.311	10.49	5.17	201.58
	StarGAN-Unfair-5	12.99	35.56	25.40	60.64	1.514	7.46	6.10	204.05
	StarGAN-Unfair-10	20.26	45.51	30.26	68.78	1.559	7.39	5.83	208.60
	StarGAN-Unfair-15	20.47	46.46	34.90	71.11	1.558	7.20	5.58	204.13
	StarGAN-Unfair-20	24.71	48.92	35.23	73.75	1.549	8.57	6.21	198.07
	StarGAN-Fair-1	0.56	3.46	4.41	20.03	1.368	7.83	3.71	228.74
	StarGAN-Fair-5	0.60	3.56	4.38	20.12	1.368	7.80	3.72	235.66
North American Birds	StarGAN-Fair-10	0.60	3.40	4.30	20.00	1.368	7.84	3.71	241.77
	StarGAN-Fair-15	0.62	3.49	4.28	20.24	1.368	7.82	3.72	228.42
	StarGAN-Fair-20	0.62	3.45	4.41	20.00	1.368	7.83	3.72	228.57
	FUNIT-1	17.07	54.11	46.72	82.36	1.364	22.18	10.04	93.03
	FUNIT-5	33.29	78.19	68.68	96.05	1.320	22.56	13.33	70.24
	FUNIT-10	37.00	82.20	72.18	97.37	1.311	22.49	14.12	67.35
	FUNIT-15	38.83	83.57	73.45	97.77	1.308	22.41	14.55	66.58
	FUNIT-20	39.10	84.39	73.69	97.96	1.307	22.54	14.82	66.14
	CycleGAN-Unfair-20	9.24	22.37	19.46	42.56	1.488	25.28	7.11	215.30
	UNIT-Unfair-20	7.01	18.31	16.66	37.14	1.417	28.28	7.57	203.83
StarGAN-Unfair-20	MUNIT-Unfair-20	23.12	41.41	38.76	62.71	1.656	24.76	9.66	198.55
	StarGAN-Fair-1	0.92	3.83	3.98	13.73	1.491	14.80	4.10	266.26
	StarGAN-Fair-5	2.54	8.94	8.82	23.98	1.574	13.84	4.21	270.12
	StarGAN-Fair-10	4.26	13.28	12.03	32.02	1.571	15.03	4.09	278.94
	StarGAN-Fair-15	3.70	11.74	12.90	31.62	1.509	18.61	5.25	252.80
	StarGAN-Fair-20	5.38	16.02	13.95	33.96	1.544	18.94	5.24	260.04
	StarGAN-Fair-1	0.24	1.17	0.97	4.84	1.423	13.73	4.83	244.65
	StarGAN-Fair-5	0.22	1.07	1.00	4.86	1.423	13.72	4.82	244.40
	StarGAN-Fair-10	0.24	1.13	1.03	4.90	1.423	13.72	4.83	244.55
	StarGAN-Fair-15	0.23	1.05	1.04	4.90	1.423	13.72	4.81	244.80
StarGAN-Fair-20	StarGAN-Fair-20	0.23	1.08	1.00	4.86	1.423	13.75	4.82	244.71
	FUNIT-1	11.17	34.38	30.86	60.19	1.342	67.17	17.16	113.53
	FUNIT-5	20.24	51.61	45.40	75.75	1.296	74.81	22.37	99.72
	FUNIT-10	22.45	54.89	48.24	77.66	1.289	75.40	23.60	98.75
	FUNIT-15	23.18	55.63	49.01	78.70	1.287	76.44	23.86	98.16
	FUNIT-20	23.50	56.37	49.81	78.89	1.286	76.42	24.00	97.94

表 1. 与公平和不公平基线的效果比较。↑意味着数字越大越好，↓意味着数字越小越好。

GAN 采用如下有条件的损失函数：

$$\mathcal{L}_{\text{GAN}}(G, D) = E_{\mathbf{x}} [-\log D^{c_x}(\mathbf{x})] + E_{\mathbf{x}, \{\mathbf{y}_1, \dots, \mathbf{y}_K\}} [\log (1 - D^{c_y}(\bar{\mathbf{x}}))] \quad (4)$$

附加到 D 的上标表示对象类，仅使用类的相应二分类预测分数来计算损失。

内容重建损失有助于 G 学习转换模型。具体地，当对输入内容图像和输入类图像使用相同图像时（在这种情况下 K = 1），损失促使 G 生成与输入相同的输出图像

$$\mathcal{L}_{\text{R}}(G) = E_{\mathbf{x}} [||\mathbf{x} - G(\mathbf{x}, \{\mathbf{x}\})||_1^1]. \quad (5)$$

特征匹配损失使训练正常化。我们首先通过从 D 重新移动最后一个（预测）层来构造一个特征提取器，称为 D_f 。然后我们使用 D_f 从转换输出 \mathbf{x} 和

类图像 $\{\mathbf{y}_1, \dots, \mathbf{y}_K\}$ 中提取特征并最小化

$$\mathcal{L}_{\text{F}}(G) = E_{\mathbf{x}, \{\mathbf{y}_1, \dots, \mathbf{y}_K\}} [D_f(\bar{\mathbf{x}})] - \sum_k \frac{D_f(\mathbf{y}_k)}{K} ||_1^1. \quad (6)$$

内容重建损失和特征匹配损失都不是图像到图像转换的新想法[29, 19, 49, 36]。我们的贡献在于将它们的使用扩展到更具挑战性和新颖性的几乎无视图的图像到图像转换设置。

4. 实验

实现。 我们设置 $\lambda_{\text{R}} = 0.1$ 和 $\lambda_{\text{F}} = 1$ 。我们使用学习率为 0.0001 的 RMSProp 优化 (3)。我们使用 GAN 损失的铰链版本[28, 32, 52, 6]和 Mescheder 等人提出的真实梯度惩罚正则化[31]。最终生成器是中间生成器[23]的历史平均版本，其中更新权重为 0.001。我们使用 K = 1 训练 FUNIT 模型，但用 K = 1, 5, 10, 15, 20 进行测试。每个训练批次包括

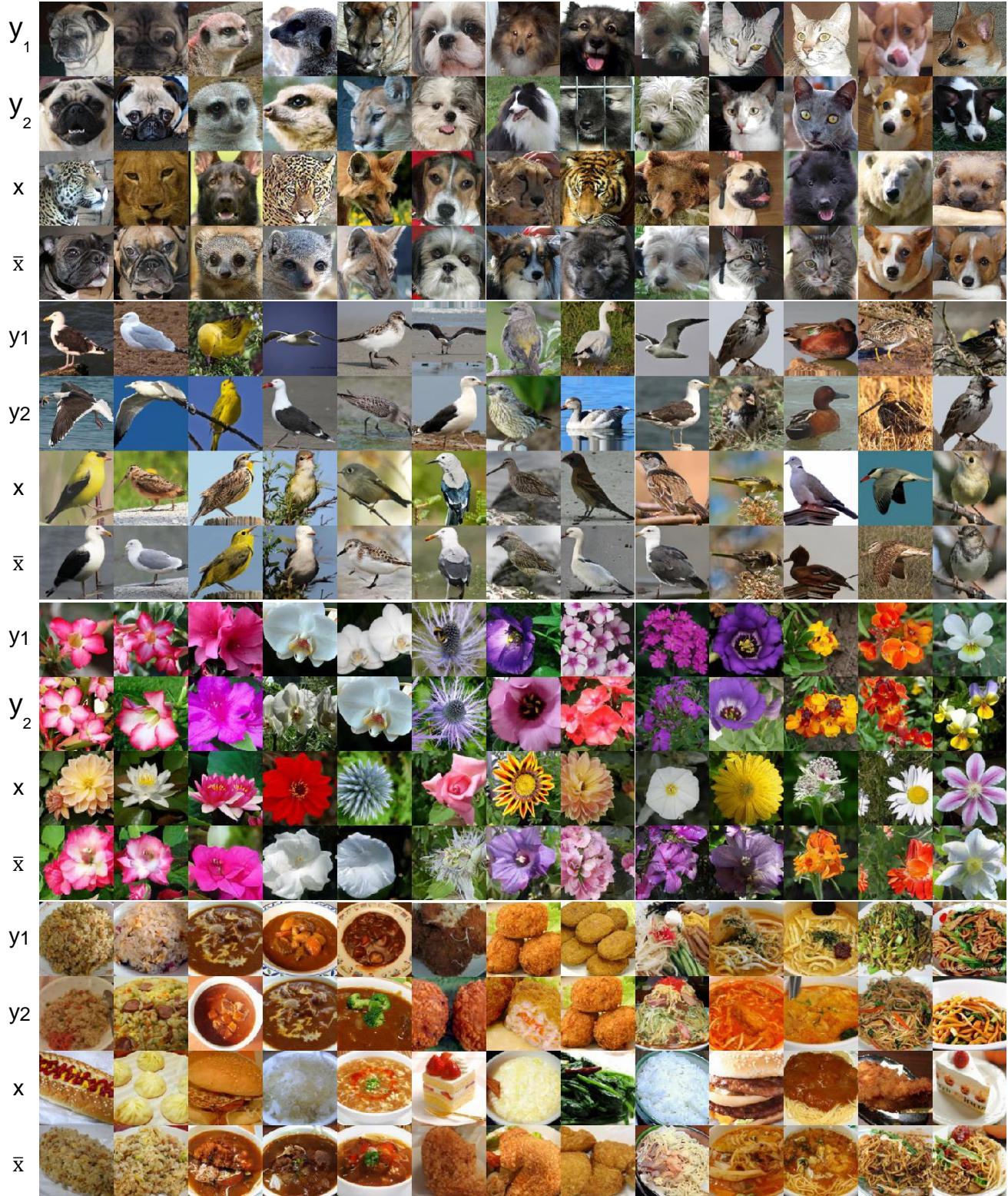


图 2. 少样本无监督的图像到图像转换结果的可视化。结果使用 FUNIT-5 模型计算。从上到下，我们得到动物面部，鸟类，花卉和食物数据集的结果。我们为每个数据集训练一个模型。对于每个示例，我们可视化 5 个随机采样的类图像 $Y_1 Y_2$ 中的 2 个，输入内容图像 x 和转换输出 \bar{x} 。结果表明，FUNIT 在困难的少样本设置下产生合理的转换输出，其中模型在训练期间没有看到来自任何目标类别的图像。我们注意到输出图像中的对象与输入具有相似的姿态。

64 个内容映像，均匀分布在 NVIDIA DGX1 机器上的 8 个 V100 GPU 上。

数据集。 我们使用以下数据集进行实验。

- **动物面孔。** 我们使用 ImageNet 中的 149 个食肉动物类的图像构建这个数据集[9]。我们首先在图像中手动标记 10000 个肉食动物面孔的边界框。然后，我们训练 Faster RCNN [13] 来检测图像中的动物面部。我们只使用具有高检测分数的边界框。这呈现出一个有 117574 个动物面孔的集合。我们将类拆分为源类集和目标类集，它们分别包含 119 和 30 个动物类。
- **鸟类[47]。** 北美洲 555 种鸟类的 48527 张图片。444 种用于源类集，111 种用于目标类集。
- **鲜花[35]。** 来自 102 种花类的 8189 张图片。源集和目标集分别有 85 种和 17 种。
- **食物[24]。** 来自 256 种食物的 31395 张图片。源集和目标集分别有 224 种和 32 种。

基线。 根据目标类的图像在训练期间是否可用，我们定义了两组基线：公平（不可用）和不公平（可用）。

• **公平。** 这是建议的 FUNIT 框架工作的设置。由于没有先前的无监督图像到图像转换方法被设计用于设置，我们通过扩展 StarGAN 方法[8]来构建基线，这是用于多级无监督图像到图像转换的现有技术。我们纯粹使用源类图像训练 StarGAN 模型。在测试期间，给定目标类别的 K 个图像，我们计算 K 图像的平均 VGG [41] Conv5 特征，并计算其与每个源类的图像的平均 VGG Conv5 特征的余弦距离。然后，我们通过将 softmax 应用于余弦距离来计算类关联向量。我们使用类关联向量作为 StarGAN 模型的输入（用 one-hot 关联向量输入代替）来生成看不见的目标类的图像。基线方法的设计理念是，类关联分数可以编码一个看不见的目标对象类与每个源类相关的方式，可以用于少样本生成。我们表示这个基线为 *StarGAN-Fair-K*。

• **不公平。** 这些基线包括训练中的目标类图像。我们将每个目标等级的可用图像 (K) 的数量从 1 到 20 改变，并训练各种无监督的图像到图像的转换模型。我们将使用 K 图像训练的 StarGAN 模型表示为每个目标类，如 *StarGAN-Unfair-K*。我们还训练了几种最先进的双域转换模型，包括 CycleGAN [54]，UNIT [29] 和 MUNIT [19]。对于他们来说，我们将源类的图像视为



图 3. 少样本图像到图像的转换性能的视觉比较。从左到右，列是输入内容图像 x ，两个输入目标类图像 y_1, y_2 ，转换来自不公平的 StarGAN 基线，来自公平 StarGAN 基线的转换结果，以及来自我们框架的结果。

第一个域和一个目标类的图像作为第二个域。这导致每个两级基线的每个数据集的 $|T|$ 无监督图像到图像转换模型。我们将这些基线标记为 *CycleGAN-Unfair-K*, *UNIT-Unfair-K* 和 *MUNIT-Unfair-K*。

对于基线方法，我们使用作者提供的源代码和默认参数设置。

评估准则。 我们使用来自源类的随机抽样的 25000 个图像作为内容图像。然后，我们通过随机抽取目标类的 K 图像将它们转换为每个目标类。这为每种竞争方法产生 $|T|$ 图像集，并用于评估。对于所有竞争方法，我们对每个内容图像使用相同的 K 图像。我们测试了一系列 K 值，包括 1, 5, 10, 15 和 20。

性能指标。 我们使用几个标准进行评估。首先，我们测量转换是否类似于目标类的图像。其次，我们检查在转换期间是否保留了类不变内容。第三，我们量化输出图像的写实照片。最后，我们测量该模型是否可用于生成目标类的图像分布。我们将在下面简要介绍这些标准的指标效果，并留下详细信息。

在附录中。

- **转换准确度** 衡量转换输出是否属于目标类。 我们使用两个 Inception-V3 [44] 分类器。 使用源类和目标类（表示为全部）训练一个分类器，而使用目标类（表示为测试）训练另一个分类器。 我们报告了 Top1 和 Top5 的准确度。
- **内容保存** 基于感知距离[22,53]的变体，称为域不变感知距离 (DIPD) [19]。 距离由两个归一化的 VGG [41] Conv5 特征之间的 L2 距离给出，这对于不同的域更具不变性[19]。
- **写实**。 这是通过初始分数 (IS) [39] 来衡量的。 我们使用训练用于测量转换准确度的两个感知分类器报告初始分数，分别用 all 和 test 表示。
- **分布匹配** 基于 Frechet' Inception Distance (FID) [17]。 我们为每个 \mathbb{T} 目标对象类计算 FID 并报告它们的平均 FID (mFID)。

主要结果。 如表 1 所示，对于 Animal Faces 和 North American Birds 数据集的所有性能指标，建议的 FUNIT 框架优于针对少样本无监督图像到图像转换任务的基线。 FUNIT 分别在动物脸部数据集上实现了 1-shot 和 5-shot 设置上的 82.36 和 96.05 的 Top-5 (测试) 准确度，在北美鸟类数据集上分别达到了 60.19 和 75.75。 它们都比相应的公平基线所取得的要好得多。 对于域不变感知距离，初始得分和 Frechet 初始距离，可以找到类似的趋势。 此外，只有 5 个样本，FUNIT 在 20-shot 设置下的表现优于所有不公平的基线。 请注意，对于 CycleGAN-Unfair-20 的结果，UNIT-Unfair-20 和 MUNIT-Unfair-20 来自 \mathbb{T} 图像到图像转换网络，而我们的方法来自单个转换网络。

该表还显示，在测试时，所提出的 FUNIT 模型的性能与可用目标图像 K 的数量正相关。 较大的 K 导致所有指标的改进，并且最大的性能提升来自 $K = 1$ 到 $K = 5$ 。 StarGAN-Fair 基线没有表现出类似的趋势。

在图 2 中，我们可视化由 FUNIT-5 计算的少样本转换结果。 结果表明，FUNIT 模型可以成功地将源类图像转换为新类的类似图像。 输入内容图像 x 中的对象的姿势和对应的输出图像 x 保持大致相同。 输出图像是逼真的，类似于目标类的图像。

在图 3 中，我们提供了视觉比较。 由于基线不是为少样本图像转换而设计的

Setting	Animal	Birds
FUNIT-5 vs. StarGAN-Fair-5	86.08	82.56
FUNIT-5 vs. StarGAN-Unfair-20	86.00	84.48
FUNIT-5 vs. CycleGAN-Unfair-20	71.68	77.76
FUNIT-5 vs. UNIT-Unfair-20	77.84	77.96
FUNIT-5 vs. MUNIT-Unfair-20	83.56	79.64

表 2. 用户偏好分数。 这些数字表明用户的年龄百分比偏好所提出的方法产生的结果而不是竞争方法产生的结果。

# of generated samples N	Animal Face		North American Birds	
	S&H [15]	FUNIT	S&H [15]	FUNIT
0	38.76		30.38	
10	40.51	42.05	31.77	33.41
50	40.24	42.22	31.66	33.64
100	40.76	42.14	32.12	34.39

表 3. 平均超过 5 次分裂的少样本分类准确度。

设置，他们在具有挑战性的转换任务中失败了它们要么生成具有大量伪像的图像，要么仅输出输入内容图像。 另一方面，FUNIT 生成高质量的图像转换输出。

用户研究。 为了比较转换输出的照片写实性和忠实度，我们使用亚马逊机械土耳其人 (AMT) 平台进行人体评估。 具体来说，我们为工人提供目标类图像和来自不同方法的两个转换输出[49,19]，并要求他们选择类似于目标类图像的输出图像。 工人有无限的时间进行选择。 我们使用 Animal Faces 和 North American Birds 数据集。 对于每次比较，我们随机生成 500 个问题，每个问题由 5 个不同的工作人员共同完成。 对于质量控制，工人必须具有超过 98% 的终身任务批准率才能参与评估。

根据表 2，人类受试者认为所提出的方法在 5-shot 设置 (FUNIT-5) 下产生的转换输出与目标级别图像相比，与在相同设置下由公平基线产生的相比更为相似 (StarGAN-Fair-5)。 即使与在训练时每个目标类别获得 20 张图像的不公平基线的结果相比，我们的转换结果仍然被认为更加忠实。

训练集中的源类数。 在图 4 中，我们使用动物数据集分析了在 1-shot 设置 (FUNIT-1) 下训练集中的性能与不同数量的源类别。 我们通过改变 69 到 119 个类的数字来绘制曲线，其间隔为 10。 如图所示，在转换精度、图像质量和分布匹配方面，性能与对象类的数量正相关。 域不变的感知距离保持平坦。 这表明 FUNIT 模型可以看到更多的对象类。

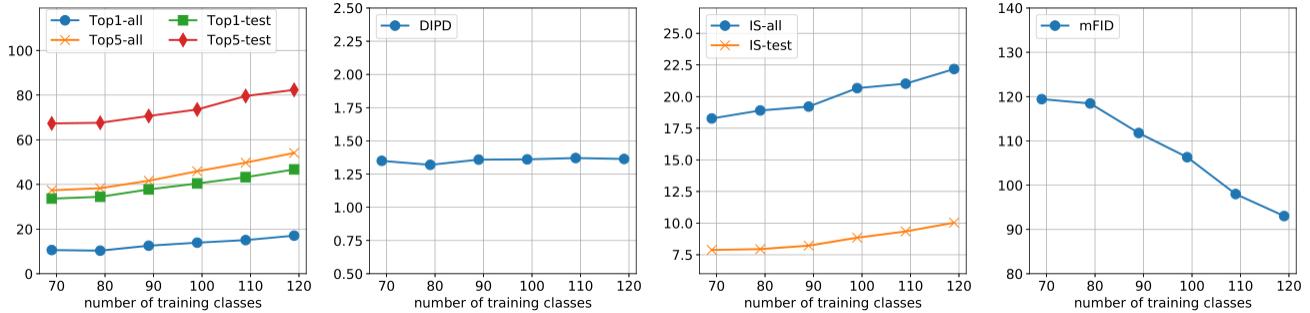


图 4. 在 Animal Faces 数据集训练期间看到的少样本图像转换性能与对象类的数量。性能与训练期间看到的源对象类的数量正相关。



图 5. 提议框架的局限性。当一个看不见的对象类的外观与源类的外观（例如花和动物的面）显著不同时。提议的 FUNIT 框架未能产生有意义的转换输出。

（训练期间更大的多样性）在测试期间表现更好。观察到鸟类数据集的类似趋势，该数据集在附录中给出。

参数分析和消融研究。我们分析了各个术语在我们的目标函数中的影响，发现它们都是必不可少的。特别是，内容重建损失换取内容保存分数的转换准确度。支持实验结果见附录。

潜插补。在附录中，我们通过保持内容编码固定，同时将类编码插入两个源类图像之间来显示插值结果。有趣的是，我们发现通过在两个源类（Siamese cat 和 Tiger）之间进行插值，我们有时可以从模型生成从未观察到的目标类（Tabby cat）。

失败的情况。附录中显示了提议算法的几个失败案例。它们包括生成混合对象，忽略输入内容图像和忽略输入类图像。

少数分类的少量转换。我们使用动物和鸟类数据集评估 FUNIT 的少样本分类。具体来说，我们使用经过训练的 FUNIT 模型为少数几个类中的每一个生成 N 个（从 1 个，50 个到 100 个不同）图像，并使用生成的图像来训练分类器。我们发现，使用 FUNIT 生成的图像训练的分类器始终比 Hariharan 等人提出的 S & H 提出的少数分类方法具有更好的性能 [15]，它基于特征幻觉，并且在样本数 N 上也具有可控变量。结果如表 3 所示，实验细节在附录中。

5. 讨论和未来工作

我们介绍了第一个少样本无监督的图像到图像转换框架。我们发现，少样本生成性能与训练期间看到的对象类别的数量正相关，并且与测试时间内提供的目标类别的数量呈正相关。

我们提供了经验证据，证明 FUNIT 可以通过利用在测试时可用的不曾见的类的几个示例图像来学习将源类的图像转换为看不见的对象类的对应图像。虽然实现了这一新功能，但 FUNIT 依赖于几个工作条件：1) 内容编码器 E_x 是否可以学习类不变的潜在代码 Z_x ，2) 类编码器 E_y 是否可以学习类特定的潜在代码 Z_y ，最重要的是，3) 类编码器 E_y 是否可以推广到看不见的对象类的图像。

我们观察到，当新类在视觉上与源类相关时，这些条件很容易满足。但是，当新对象类的外观与源类的外观显著不同时，FUNIT 无法实现如图 5 所示的转换。在这种情况下，FUNIT 倾向于生成输入内容图像的颜色改变版本。这是不可取的，但可以理解，因为外观分布发生了巨大变化。解决这个限制是我们未来的工作。

参考文献

- [1] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, and A. Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. arXiv preprint arXiv:1802.10151, 2018. 2
- [2] A. Anoosheh, E. Agustsson, R. Timofte, and L. Van Gool. Combogan: Unrestrained scalability for image domain trans-lation. arXiv preprint arXiv:1712.06909, 2017. 2
- [3] S. Benaim and L. Wolf. One-sided unsupervised domain mapping. In Advances in Neural Information Processing Systems (NIPS), 2017. 2
- [4] S. Benaim and L. Wolf. One-shot unsupervised cross domain translation. In Advances in Neural Information Processing Systems (NIPS), 2018. 2
- [5] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 2
- [6] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. In International Conference on Learning Representations (ICLR), 2019. 4
- [7] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In IEEE International Conference on Computer Vision (ICCV), 2017. 11
- [8] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 2, 6
- [9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009. 4
- [10] M. Dixit, R. Kwitt, M. Niethammer, and N. Vasconcelos. Aga: Attribute-guided augmentation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 2
- [11] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2006. 2
- [12] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In International Conference on Machine Learning (ICML), 2017. 2
- [13] R. Girshick. Fast r-cnn. In Advances in Neural Information Processing Systems (NIPS), 2015. 6
- [14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Gen-erative adversarial nets. In Advances in Neural Information Processing Systems (NIPS), 2014. 1
- [15] B. Hariharan and R. B. Girshick. Low-shot visual recogni-tion by shrinking and hallucinating features. In IEEE Inter-national Conference on Computer Vision (ICCV), 2017. 2, 7, 8, 14, 16
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 3
- [17] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In Advances in Neural Information Processing Systems (NIPS), 2017. 7, 12
- [18] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In IEEE Inter-national Conference on Computer Vision (ICCV), 2017. 3, 11
- [19] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. European Conference on Computer Vision (ECCV), 2018. 2, 3, 4, 6, 7, 11
- [20] L. Hui, X. Li, J. Chen, H. He, and J. Yang. Unsuper-vised multi-domain image translation with domain-specific encoders/decoders. arXiv preprint arXiv:1712.02050, 2017. 2
- [21] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In IEEE Conference on Computer Vision and Pattern Recogni-tion (CVPR), 2017. 11
- [22] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision (ECCV), 2016. 3, 7, 11
- [23] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and varia-tion. In International Conference on Learning Representa-tions (ICLR), 2018. 4
- [24] Y. Kawano and K. Yanai. Automatic expansion of a food im-age dataset leveraging existing categories with domain adap-tation. In European Conference on Computer Vision (ECCV) Workshop, 2014. 6
- [25] T. Kim, M. Cha, H. Kim, J. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In International Conference on Machine Learning (ICML), 2017. 1, 2
- [26] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. K. Singh, and M.-H. Yang. Diverse image-to-image translation via disentangled representation. In European Conference on Computer Vision (ECCV), 2018. 2
- [27] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou. Revisiting batch normalization for practical domain adaptation. arXiv preprint arXiv:1603.04779, 2016. 11
- [28] J. H. Lim and J. C. Ye. Geometric gan. arXiv preprint arXiv:1705.02894, 2017. 4
- [29] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In Advances in Neural Information Processing Systems (NIPS), 2017. 1, 2, 3, 4, 6
- [30] M.-Y. Liu and O. Tuzel. Coupled generative adversarial net-works. In Advances in Neural Information Processing Sys-tems (NIPS), 2016. 1, 2
- [31] L. Mescheder, A. Geiger, and S. Nowozin. Which training methods for gans do actually converge? In International Conference on Machine Learning (ICML), 2018. 4, 11
- [32] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spec-tral normalization for generative adversarial networks. In In-ternational Conference on Learning Representations (ICLR), 2018. 4
- [33] T. Munkhdalai and H. Yu. Meta networks. In International Conference on Machine Learning (ICML), 2017. 2

- [34] A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999, 2018. 2
- [35] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In Indian Conference on Computer Vision, Graphics & Image Processing, 2008. 6
- [36] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially adaptive normalization. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 4
- [37] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In International Conference on Learning Representations (ICLR), 2017. 2
- [38] R. Salakhutdinov, J. Tenenbaum, and A. Torralba. One-shot learning with a hierarchical nonparametric bayesian model. In International Conference on Machine Learning (ICML) Workshop, 2012. 2
- [39] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In Advances in Neural Information Processing Systems (NIPS), 2016. 7, 11, 12
- [40] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 2
- [41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations (ICLR), 2015. 6, 7, 11
- [42] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In Advances in Neural Information Processing Systems (NIPS), 2017. 2
- [43] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 2
- [44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 6, 11, 12
- [45] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. In International Conference on Learning Representations (ICLR), 2017. 1, 2
- [46] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 11
- [47] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. 6
- [48] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In Advances in Neural Information Processing Systems (NIPS), 2016. 2
- [49] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 4, 7, 11
- [50] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan. Low-shot learning from imaginary data. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 2
- [51] Z. Yi, H. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In IEEE International Conference on Computer Vision (ICCV), 2017. 1, 2
- [52] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. arXiv preprint arXiv:1805.08318, 2018. 4
- [53] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 7, 11
- [54] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In IEEE International Conference on Computer Vision (ICCV), 2017. 1, 2, 3, 6
- [55] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In Advances in Neural Information Processing Systems (NIPS), 2017. 2

A. 网络架构

在本节中，我们首先讨论少样本图像转换器的体系结构，然后介绍多任务对抗判别器的体系结构。已发布的代码中提供了其他详细信息。

A.1. 少样本图像转换器架构

少样本图像转换器由三个子网络组成：内容编码器，类编码器和解码器，如图 6 所示。内容编码器将输入内容图像映射到内容潜在编码，这是一个特征图。如果输入图像的分辨率为 128x128，则特征图的分辨率为 16x16，因为有 3 个步幅为 2 的下采样操作。此功能映射旨在编码类不变的内容信息。它应编码区域的位置，而不是其特定类别的外观（见底部¹）。另一方面，类编码器将一组 K 类图像映射到类潜在编码，该类潜在编码是向量并且旨在是特定类的。它首先使用类似 VGG 的网络将每个输入类图像映射到中间潜在编码。然后对这些潜在向量进行元素平均以产生最终类潜在编码。

如在图中所示，解码器首先将类特定潜码解码为一组均值和方差矢量 (μ_i, σ_i^2)，其中 $i=1, 2$ 。然后将这些矢量用作 AdaIN 残差块中的仿射变换参数，其中 σ_i^2 们是缩放因子， μ_i 们是偏差。对于每个残余块，将相同的仿射变换应用于特征图中的每个空间位置。它控制内容潜在编码如何解码为输出图像。

实现。 图 6 中每个块中显示的数字表示层中的卷积核数量。网络中包含的非线性和正则化操作在可视化中被排除，以避免混乱的表示。对于内容编码器，每个层后面都是实例正则化和 ReLU 非线性。对于类编码器，每层后面都是 ReLU 非线性。对于解码器，除了 AdaIN 残余块之外，每层都跟着实例正则化和 ReLU 非线性。我们使用最近邻居上采样将每个空间维度上的特征映射放大 2 倍。

A.2. 判别器架构

我们的判别器是 Patch GAN 判别器[21]。它利用 Leaky ReLU 非线性并且不采用标准化。判别器由一个卷积层组成，然后是 10 个激活第一个残余块[31]。该体系结构通过以下说明

¹ 例如，在动物面部转换任务中，它应该编码耳朵的位置而不是它们的形状和颜色。

运行架构链： $Conv-64 \rightarrow ResBlk-128 \rightarrow ResBlk-128 \rightarrow AvePool2x2 \rightarrow ResBlk-256 \rightarrow ResBlk-256 \rightarrow AvePool2x2 \rightarrow ResBlk-512 \rightarrow ResBlk-512 \rightarrow AvePool2x2 \rightarrow ResBlk-1024 \rightarrow ResBlk-1024 \rightarrow AvePool2x2 \rightarrow ResBlk-1024 \rightarrow ResBlk-1024 \rightarrow Conv-||S||$ 其中 $||S||$ 是源类的数量。

B. 额外实验

在本节中，我们将首先详细讨论我们的评估标准。然后，我们将提出主要论文中提到的其他实验结果。

B.1. 有关性能指标的其他详细信

转换准确性。 我们使用两个 Inception-V3 [44] 分类器来测量转换精度。表示为 all 的第一个分类是通过微调 ImageNet 预训练的 Inception-V3 模型获得的，该任务是对所有源和目标对象类进行分类（例如，Animal Faces 数据集的所有 149 个类以及所有 555 个北美鸟类数据集类别。第二个分类器（表示为测试）是通过在对目标对象类进行分类的任务上微调 ImageNet 预训练的 Inception-V3 模型获得的（例如，动物面部数据集的 30 个目标类和北美鸟类数据集的 111 个目标类）。我们将分类器应用于转换输出，以查看它们是否可以将输出识别为目标类的图像。如果是，我们将其表示为正确的转换。我们使用 Top1 和 Top5 精度比较竞争模型的性能。因此，我们有 4 个转换准确度评估指标：Top1-all, Top5-all, Top1-test 和 Top5-test。具有更高精度的无监督图像到图像转换模型更好。我们注意到类似的评估准则用于比较语义标签图上的图像到图像转换模型和图像转换任务[21,49,7]。

内容保存。 我们使用域不变的感知距离 (DIPD) 来量化内容保留性能[19]。DIPD 是感知距离的变体[22,53]。为了计算 DIPD，我们首先从输入内容图像以及输出转换图像中提取 VGG [41] conv5 特征。然后，我们将实例标准化[46] 应用于要素，这将删除它们的均值和方差。这样，我们可以在特征[18,27]中过滤掉许多特定于类的信息，并专注于类不变的相似性。DIPD 由实例规范化特征之间的 L2 距离给出。

写实。 我们使用初始分数 (IS) [39]，它被广泛用于量化图像生成性能。设 $p(t|y)$ 是输出转换图像 y 上的初始模型的类标签 t 的分布。

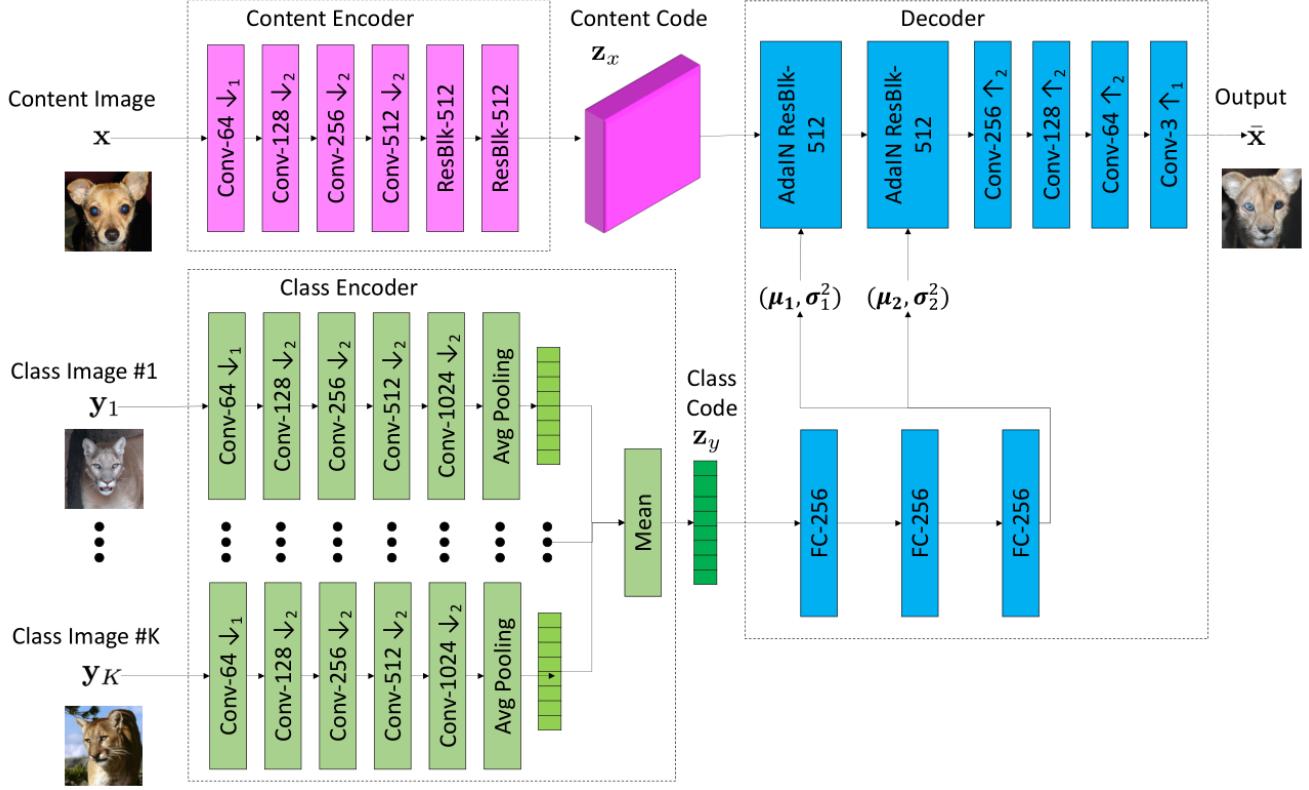


图 6.生成器体系结构的可视化。为了生成转换输出 \bar{x} ，转换器组合从类图像 y_1, \dots, y_K 中提取的类潜在代码 z_y ，从输入内容图像中提取内容潜在代码 z_x 。请注意，非线性和规范化操作不包含在可视化中。

初始分数由下式给出：

$$\text{IS}_C = \exp(\mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})}[\text{KL}(p(t|\mathbf{y})|p(t))]) \quad (7)$$

其中 $p(t) = \int_{\mathbf{y}} p(t|\mathbf{y})d\mathbf{y}$ 。在 Salimans 等人的论证中 [39] 那个初始分数与神经网络生成图像的视觉质量正相关。

分配匹配。 Frechet 初始距离 FID [17] 设计用于测量两组图像之间的相似性。我们使用来自 ImageNet 预训练的 Inception-V3 [44] 模型的最后一个平均合并层的激活作为用于计算 FID 的图像的特征向量。由于我们有 $|\mathcal{T}|$ 看不见的类，我们将源图像转换为每个 $|\mathcal{T}|$ 看不见的类，并生成 $|\mathcal{T}|$ 组转换输出。对于每组 $|\mathcal{T}|$ 平移输出，我们计算该组与相应的完全真实图像集之间的 FID。这将呈现 $|\mathcal{T}|$ FID 分数。 $|\mathcal{T}|$ FID 分数的平均值用作我们的最终分布匹配性能度量，其被称为平均 FID (mFID)。

B.2. 训练集中的源类数

在主要论文中，我们表明，少数转换表现与动物面部转换任务训练集中的源类数量正相关。

在图 7 中，我们展示了鸟类转换任务的相同情况。具体而言，我们使用北美鸟类数据集报告了所提出模型的性能与训练集中可用源类别的数量。我们将源类的数量从 189,222,289,333,389 变为 444，并绘制性能分数。我们发现得分的曲线遵循与主要论文中显示的 Animal Faces 数据集相同的趋势。当模型在训练期间看到大量源类时，它在测试期间表现更好。

B.3. 训练迭代与性能

在图 8 中，我们在一次性设置 (UNIT-1) 上绘制了所提出的模型相对于训练迭代的性能。转换准确性、内容保存、图像质量和分布匹配分数通常会随着更多的操作而提高。这种改进在早期阶段更为显著，并且在 10000 次迭代中减缓。因此，我们使用 10000 次迭代作为整个论文中重新移植实验结果的默认参数。

B.4. 参数灵敏度和消融研究

在表 4 中，我们分析了权重值对所提出模型的内容图像重建损失的影响

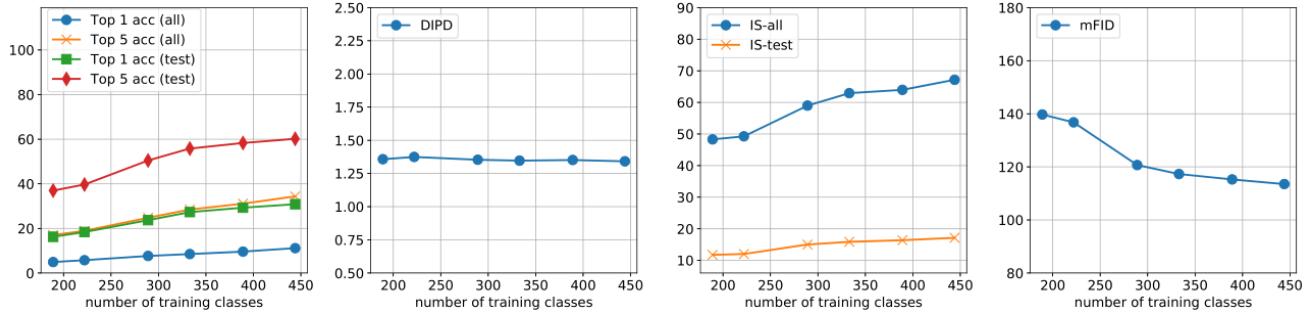


图 7. 在北美鸟类数据集训练期间看到的少样本图像转换性能与对象类别数量。

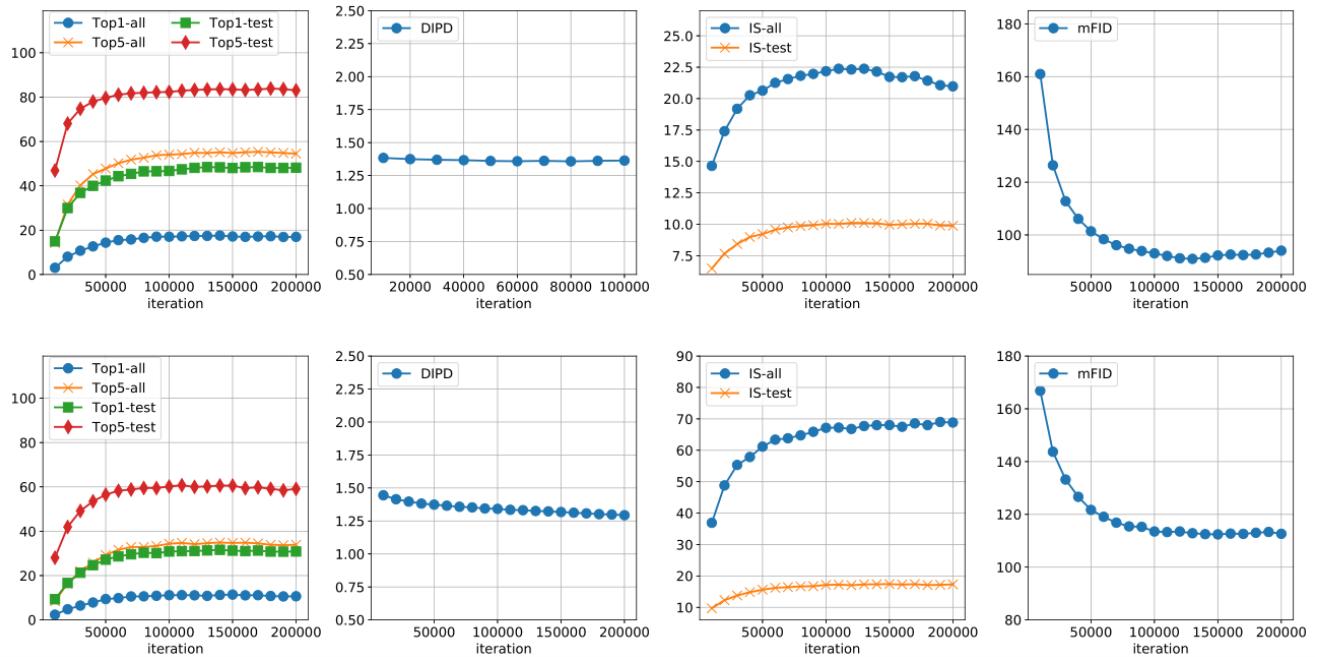


图 8. 少样本图像转换性能与训练迭代次数。第一行: Animal Faces 数据集上的结果; 底行: 北美鸟类数据集的结果。

Setting	Top1-all ↑	Top5-all ↑	Top1-test ↑	Top5-test ↑	DIPD ↓	IS-all ↑	IS-test ↑	mFID ↓
$\lambda_R = 0.01$	16.02	52.30	45.52	81.68	1.370	21.80	9.73	94.98
$\lambda_R = 0.1$	17.07	54.11	46.72	82.36	1.364	22.18	10.04	93.03
$\lambda_R = 1$	16.60	52.05	45.62	81.77	1.346	22.21	9.81	94.23
$\lambda_R = 10$	13.04	44.32	39.06	75.81	1.298	20.48	8.90	108.71

表 4. 内容图像重建损失权重的参数灵敏度分析 λ_R 。↑ 意味着数字越大越好，↓ 意味着数字越小越好。值 0.1 提供了内容保存和转换准确性之间的良好平衡，也被用作整篇论文的默认值。我们使用 FUNIT-1 模型进行本实验。

Setting	Top1-all ↑	Top5-all ↑	Top1-test ↑	Top5-test ↑	DIPD ↓	IS-all ↑	IS-test ↑	mFID ↓
FM	15.33	52.98	46.33	82.43	1.401	22.45	9.86	92.98
GP	1.15	4.74	3.18	15.50	1.752	1.78	1.84	316.56
proposed	17.07	54.11	46.72	82.36	1.364	22.18	10.04	93.03

表 5. 对象术语的消融研究。↑ 意味着数字越大越好，↓ 意味着数字越小越好。FM 表示已删除功能匹配丢失项的建议框架的设置，而 GP 表示拟议框架的设置，没有梯度惩罚损失。默认设置在大多数情况下在各种标准下呈现更好的表现。我们在此实验中使用 FUNIT-1 模型。



图 9. 失败案例。所提出的 FUNIT 模型的典型故障情况包括生成混合对象（例如，列 1,2,3 和 4），忽略输入内容图像（例如，列 5 和 6），以及忽略输入类图像（例如，列 7）。

在使用 Animal Faces 数据集时。我们发现较大的 R 值导致较小的域不变的感知距离，并且降低了转换精度。该表显示 $R = 0.1$ 提供了良好的权衡，我们在整篇论文中将其用作默认值。有趣的是， $R = 0.01$ 的非常小的权重值导致内容保存和转换准确性的性能下降。这表明内容重建损失有助于规范训练。

表 5 显示了消融研究的结果，该研究使用 Animal Faces 数据集研究了所提出的算法中各种损失成分的影响。我们发现删除特征匹配损失项会导致性能略有下降。但是当消除零中心梯度惩罚时，内容保留和转换准确度都会大幅降低。

B.5. 失败案例

图 9 说明了所提算法的几种失败情况。它们包括生成混合对象，忽略输入内容图像，以及忽略输入类图像。

B.6. 潜插补

我们探索了类编码器学到的潜在空间。在图 10 中，我们使用 t-SNE 在二维空间中可视化类代码。可以看出，来自类似类的图像在类嵌入空间中被组合在一起。

图 11 显示了通过在两个源类图像之间插入类编码时保持内容编码固定的插值结果。有趣的是，我们发现通过在两个源类 (Siamese cat 和 Tiger) 之间进行插值，我们有时可以从模型生成从未观察到的目标类 (Tabby cat)。这表明类编码器学习了一般类特定的表示，从而能够推广到新的类。

B.7. 少数分类的少量转换

正如主要论文中所提到的，我们使用 FUNIT 生成器生成的图像进行实验，以便在 1-shot 设置中训练新类别的分类器，使用 Animal Faces 和 North American Birds 数据集。在 Hariharan 等人的设置之后 [15]，我们创建了 5 个不同的 1-shot 训练分组，每个分组都有训练，验证和测试集。训练集由 $|\mathbb{T}|$ 图像组成，每个 $|\mathbb{T}|$ 测试类都有一个图像。验证集包含来自每个测试类的 20-100 个图像。测试集包含剩余的测试类图像。

我们使用 FUNIT 生成器通过使用分类训练集中的图像作为类图像输入并且从源类中随机采样的图像作为内容图像输入来生成合成训练集。我们使用原始和合成训练集训练分类器。我们将我们的方法与 Hariharan 等人的 Shrink and Hallucinate (S & H) 方法进行比较 [15]，学习生成对应于新类的最终层特征。我们使用预训练的 10 层 ResNet 网络作为特征提取器，它纯粹使用源类图像进行预训练，并在目标类上训练线性分类器。我们发现，对生成图像的损失加权低于真实图像的重要性。我们使用验证集对重量值和重量衰减值进行详尽的网格搜索，并在测试集上报告性能。为了公平比较，我们还对 S & H 方法执行相同的详尽搜索。

在主要论文的表 4 中，我们报告了我们的方法和 S & H 方法 [15] 在这两个具有挑战性的细粒度分类任务的不同数量的生成样本（即 FUNIT 的图像和 S & H 的特征）上的性能。两种方法都比基线分类器表现更好，基线分类器仅使用每个小类提供的单个实际图像。使用我们生成的图像，我们获得了比生成特征的 S & H 方法大约 2% 的改进。

基础 10 层 ResNet 网络训练了 90 个时期，初始学习率为 0.1，每 30 个时期由 10 个因子衰减。线性分类器在新类别上的重量衰减选自 15 个对数间隔值，在 0.000001 和 0.1 之间并包括 0.000001 和 0.1。生成的图像和特征上的损失的损失乘数是从 7 和 0.001 之间的对数间隔值中选择的。重量衰减和损失乘数的值是基于在 Split #1 上训练时获得的最佳验证集精度来选择的。然后固定这些值并用于所有剩余的分裂 2-5。使用固定特征学习 L2 正则化分类器的任务是凸优化问题，并且我们使用 L-BFGS 算法的线搜索，因此不必指定学习速率。

在表 6 和表 7 中，我们报告了测试精度及其关于



图 10. 使用 t-SNE 对 50 个源类的 5000 个图像进行类代码的二维表示。请放大以了解详情。



图 11. 通过在源类的两个类代码之间进行插值时保持内容代码固定来进行插值。

动物面部和北美鸟类数据集的所有 5 个 1-shot 分割的 1-shot 学习的相关方差。在所有实验中，我们只学习一个新分类器

层使用从用于训练图像生成器的类集训练的网络中提取的特征。

Method	# of generated Samples	Split					Average Accuracy
		1	2	3	4	5	
Baseline	0	38.81 \pm 0.01	41.99 \pm 0.03	39.13 \pm 0.01	37.05 \pm 0.02	36.82 \pm 0.01	38.76
FUNIT	10	41.20 \pm 0.41	46.25 \pm 0.27	42.65 \pm 0.41	40.75 \pm 0.20	39.39 \pm 0.31	42.05
	50	41.24 \pm 0.16	46.27 \pm 0.07	43.15 \pm 0.06	41.01 \pm 0.19	39.43 \pm 0.09	42.22
	100	41.01 \pm 0.18	46.72 \pm 0.05	42.89 \pm 0.09	40.73 \pm 0.20	39.33 \pm 0.04	42.14
S&H [15]	10	39.87 \pm 0.47	42.69 \pm 0.34	41.42 \pm 0.39	39.95 \pm 0.58	38.64 \pm 0.42	40.51
	50	39.93 \pm 0.15	42.62 \pm 0.28	40.89 \pm 0.09	39.31 \pm 0.17	38.44 \pm 0.13	40.24
	100	40.05 \pm 0.31	41.72 \pm 0.19	41.29 \pm 0.16	41.33 \pm 0.21	39.39 \pm 0.16	40.76

表 6. 当使用生成的图像和 1 个真实图像时, Animal Faces 数据集的 5 个分割的单次精确度。每次分割报告超过 5 次独立运行的平均准确度(每次采样不同的生成图像集)。

Method	# of generated Samples	Split					Average Accuracy
		1	2	3	4	5	
Baseline	0	30.71 \pm 0.02	29.04 \pm 0.01	31.93 \pm 0.01	29.59 \pm 0.01	30.64 \pm 0.02	30.38
FUNIT	10	32.94 \pm 0.49	33.29 \pm 0.25	35.15 \pm 0.22	31.20 \pm 0.20	34.48 \pm 0.58	33.41
	50	32.92 \pm 0.34	33.78 \pm 0.25	35.04 \pm 0.10	31.80 \pm 0.14	34.66 \pm 0.17	33.64
	100	33.83 \pm 0.16	33.99 \pm 0.12	36.05 \pm 0.14	32.01 \pm 0.10	36.09 \pm 0.19	34.39
S&H [15]	10	30.55 \pm 0.11	31.96 \pm 0.30	34.18 \pm 0.19	30.65 \pm 0.09	31.49 \pm 0.24	31.77
	50	31.39 \pm 0.07	30.59 \pm 0.11	33.60 \pm 0.05	30.92 \pm 0.20	31.81 \pm 0.18	31.66
	100	30.83 \pm 0.10	32.03 \pm 0.09	34.39 \pm 0.17	31.12 \pm 0.10	32.23 \pm 0.15	32.12

表 7. 使用生成的图像和 1 个真实图像时, North Amercian Birds 数据集的 5 个分割的单次精确度。每次分割报告超过 5 次独立运行的平均准确度(每次采样不同的生成图像集)。

B.8. 转换结果的附加可视化

在图 12,13 和 14 中, 我们显示了动物面部图像转换任务,

鸟类图像转换任务, 花卉图像转换任务和食物图像转换任务

的附加少样本转换结果。结果使用 FUNIT-5 计算。



图 12. 少样本动物面部图像转换任务的附加可视化结果。所有结果均使用相同的 FUNIT-5 模型计算。通过访问来自目标类的 5 个图像，可以重新使用该模型以在测试时间中生成动态指定的目标类的图像。变量 x 是输入内容图像， y_1 和 y_2 是 5 个输入目标类图像中的 2 个， \bar{x} 是转换输出。我们发现转换输出中的动物面部与输入内容图像具有相似的姿势，但外观类似于类图像中动物面部的外观。



图 13. 少样本鸟类图像转换任务的附加可视化结果。所有结果均使用相同的 FUNIT-5 模型计算。通过访问来自目标类的 5 个图像，可以重新使用该模型以在测试时间中生成动态指定的目标类的图像。变量 x 是输入内容图像， y_1 和 y_1 是 5 个输入目标类图像中的 2 个， \bar{x} 是转换输出。我们发现转换输出中的鸟与输入内容图像具有相似的姿势，但外观类似于类图像中鸟类的外观。

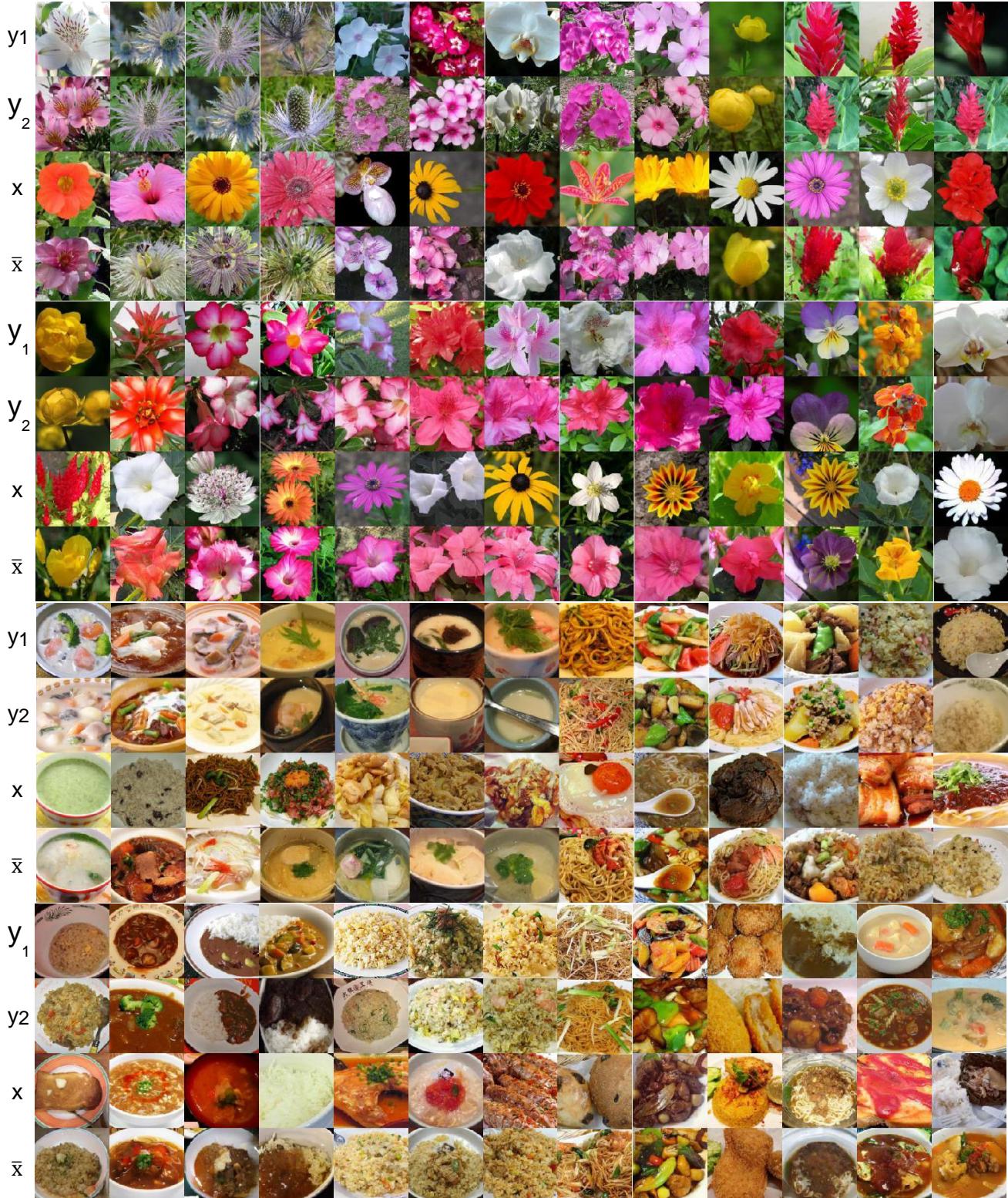


图 14.关于少量花卉和食物图像转换任务的附加可视化结果。 使用相同的 FUNIT-5 模型计算相同任务的所有结果。 通过访问来自目标类的 5 个图像，可以重新使用该模型以在测试时间中生成动态指定的目标类的图像。 变量 x 是输入内容图像， y_1 和 y_1 是 5 个输入目标类图像中的 2 个， \bar{x} 是转换输出。 对于花卉转换，我们发现输出中的花朵和输入图像具有相似的姿势。 对于食物转换，碗和盘保持在相同位置，同时食物从一种变为另一种。