The Unreasonable Effectiveness of Deep Features as a Perceptual Metric

Dezeming Family

2023 年 3 月 10 日

DezemingFamily 系列文章和电子书**全部都有免费公开的电子版**，可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列电子书，可以从我们的网站 [https://dezeming.top/] 找到最新的版本。对文章的内容建议和出现的错误也欢迎在网站留言。

目录

一　论文引文

本文涉及很多数据集和描述，虽然非常简单，但我还是希望能详细地翻译和整合一下本文，因为这篇文章确实比较重要，该评价方法也确实在现在使用比较广泛。

虽然人类几乎不费吹灰之力就能快速评估两幅图像之间的感知相似性，但其潜在过程被认为相当复杂。尽管如此，当今最广泛使用的感知度量，如 PSNR 和 SSIM，都是很简单的函数，无法解释人类感知的许多细微差别。最近，深度学习论坛发现，基于 ImageNet 分类训练的 VGG 网络的特征作为图像合成的训练损失非常有用。但这些所谓的“知觉损失”有多感性？哪些要素对它们的成功至关重要？为了回答这些问题，我们引入了一个新的人类感知相似性判断数据集。我们系统地评估不同架构和任务的深层特征，并将其与经典度量进行比较。我们发现，在我们的数据集上，深度特征在很大程度上优于所有以前的指标。更令人惊讶的是，这一结果并不局限于 ImageNet 训练的 VGG 功能，而是适用于不同的深度架构和监督级别（有监督、自我监督甚至无监督）。我们的结果表明，感知相似性是一种在深度视觉表征中共享的新兴属性。

比较数据项的能力可能是所有计算基础上最基本的操作。在计算机科学的许多领域，它并没有带来太大的困难：人们可以使用 Hamming 距离来比较 binary patterns；用 edit distance 距离来比较文本文件；用 Euclidean distance 来比较向量等。视觉模式不仅是非常高维和高度相关，而且视觉相似性的概念往往是主观的，旨在模仿人类的视觉感知。例如，在图像压缩中，目标是人类观察者无法将压缩图像与原始图像区分开来，而不管它们的像素表示可能非常不同。

一些度量方法，比如 $\updownarrow_2$ Euclidean distance 以及相关的 PSNR，用来评估图像相似性都是不足的，比如对一幅图像进行模糊，它可能与原图有很小的 $\updownarrow_2$ 值，但是视觉上已经相差了很多。如何让感知距离（perceptual distance，即感知上两幅图像的相似度）更接近，已经有了不少数值方法，比如 SSIM、MSSIM、FSIM 和 HDR-VDP。然而，构建一个符合人类的感知度量也是不太容易的，因为：

- 人类的感知相似性取决于高阶图像结构
- 人类的感知相似性与图像内容是相关的
- 人类的感知相似性可能实际上不构成距离度量

与图像内容相关的关键在于，我们可以同时记住许多不同的“相似感”：红色圆圈更类似于红色正方形还是蓝色圆圈？由于判断的上下文依赖性和成对性（比较两个图像之间的相似性），将函数直接拟合到人类判断可能很难。事实上，我们在本文中提到了一个负面结果，即即使在包含许多失真类型的大规模数据集上进行训练时，这种方法也无法推广。

相反，是否有一种方法可以在不直接训练的情况下学习感知相似性的概念？计算机视觉界已经发现，深度卷积网络的内部激活，尽管在高级图像分类任务上进行了训练，但作为更广泛的任务的代表空间，通常非常有用。例如，VGG 架构中的特征已用于神经风格转移、图像超分辨率重建和条件图像合成等任务。这些方法测量 VGG 特征空间中的距离，作为图像回归问题的“感知损失”。但这些“知觉损失”实际上与人类的视觉感知相符吗？它们与传统的感知图像评估指标相比如何？网络架构是否重要？它是否必须接受 ImageNet 分类任务的训练，或者其他任务是否同样有效？网络是否需要训练？

在本文中，我们在一个新的大规模人类判断数据库上评估了这些问题，并得出了几个令人惊讶的结论。我们发现，为高级分类任务训练的网络的内部激活，即使是跨网络架构并且没有进一步的校准，确实与人类的感知判断相对应。事实上，它们比 SSIM 和 FSIM 这样的常用度量要好得多，这些度量不是为了处理空间模糊是一个因素的情况而设计的 [48]。

此外，性能最好的自监督网络，包括 BiGAN、跨信道预测 (cross-channel prediction) 和解谜 (puzzle solving)，在这项任务中表现同样出色，即使没有人类标记的训练数据的好处。即使是简单的叠加了 k 均值初始化的无监督网络，也大大超过了经典指标。这说明了跨网络，甚至跨架构和训练信号共享的一种新兴属性。然而，重要的是，具有一些训练信号似乎至关重要——随机初始化的网络实现的性能要低得多。

我们的研究基于新收集的感知相似性数据集，使用了大量失真和真实算法输出。它既包含传统失真，如对比度和饱和度调整、噪声模式、滤波和空间扭曲操作，也包含基于 CNN 的算法输出，如自动编码、去噪和彩色化，这些都是由各种架构和损耗产生的。我们的数据集比以前的此类数据集更丰富、更多样。我

们还收集了超分辨率、帧插值和图像去模糊任务的真实算法输出的判断，这一点特别重要，因为这些是感知度量的真实使用案例。

我们表明，通过学习层激活的简单线性缩放，我们的数据可以用于“校准”现有网络，以更好地匹配低级人类判断。我们的结果与这样一个假设一致：即感知相似性本身并不是一种特殊的功能，而是一种被调整为可以预测真实世界的重要结构的视觉表征的结果。在语义预测任务中的有效的表示，也是欧几里德距离高度预测感知相似性判断的表示。

- 我们引入了一个大规模、多样的感知相似性数据集，包含 $484k$ 个人类判断。我们的数据集不仅包括参数化失真，还包括真实算法 (real algorithm) 输出（比如超分辨重建算法的输出）。
- 我们还收集了不同知觉测试的判断：显著差异（just noticeable differences, JND）。我们表明，在监督、自我监督和无监督目标上训练的深度特征对低级感知相似性的建模出奇地好，优于以前广泛使用的度量。
- 我们证明，网络架构本身并不能解释性能：未经训练的网络的性能要低得多。
- 利用我们的数据，我们可以通过“校准 (calibrating)”来自预训练网络的特征响应来提高性能。

二　数据集描述

数据集是本文 [1] 中的一个非常重要的方面，我们也会详细描述一下。

为了评估不同感知度量的性能，我们使用两种方法收集了一个大规模的高度多样化的感知判断数据集。我们的主要数据收集采用了二选一强制选择 (two alternative forced choice（2AFC）) 测试，该测试询问两种失真中的哪一种更类似于参考图像。第二个实验验证了这一点，我们在这里进行了一个显著差异 (just noticeable difference （JND）) 测试，该测试询问两个片段（一个参考片段和一个失真片段）是否相同或不同。这些判断是在广泛的失真空间（用各种失真算法得到的失真图像，比如图像模糊算法）和真实算法输出（比如超分辨重建算法生成的图像）上收集的。

2 1　图像失真

2 1.1　Traditional distortions

下表中有许多传统失真方法，我们的传统失真是由基本的基础图像编辑操作执行的，包括相机测量失真 (photometric distortions)、随机噪声 (random noise)、模糊 (blurring)、空间偏移和损坏 (spatial shifts and corruptions) 以及压缩伪影 (compression artifacts)。

Sub-type	Distortion type
Photometric	lightness shift, color shift, contrast, saturation
Noise	uniform white noise, Gaussian white, pink, & blue noise, Gaussian colored (between violet and brown) noise, checkerboard artifact
Blur	Gaussian, bilateral filtering
Spatial	shifting, affine warp, homography, linear warping, cubic warping, ghosting, chromatic aberration,
Compression	jpeg

失真程度的严重性可以被参数化，例如，对于高斯模糊，核宽度决定了输入图像的损坏量。我们还按顺序组合成对的失真，以增加可能失真的总体空间。总共有 20 个失真和 308 个顺序合成的失真。

2 1.2　CNN-based distortions

为了更精确地模拟基于深度学习的方法可能产生的 artifacts，我们创建了一组由神经网络生成的失真。我们通过探索各种任务、架构和损失来模拟可能的算法输出，如下表所示。这些任务包括自动编码、去噪、彩色化和超分辨率。所有这些任务都可以通过对输入应用适当的损坏 (corruption) 来实现。

Parameter type	Parameters
Input corruption	null, pink noise, white noise, color removal, downsampling
Generator network architecture	# layers, # skip connections, # layers with dropout, force skip connection at highest layer, upsampling method, normalization method, first layer stride # channels in 1^{st} layer, max # channels
Discriminator	number of layers
Loss/Learning	weighting on oixel-wise (ℓ_1), VGG, discriminator losses, learning rate

总共，我们生成了 96 个“去噪自动编码器”，并将其用作基于 CNN 的失真函数。我们在 $1.3M$ ImageNet 数据集上训练这些网络 1 epoch。每个网络的目标不是解决任务本身，而是探索困扰基于深度学习的方法输出的常见 artifacts。

我们基于 CNN 的失真是由随机变化的参数形成的，如任务、网络架构和学习参数。失真的目标是模拟真实算法输出中的合理失真。

2 1.3 Distorted image patches from real algorithms

图像评估算法的真正测试是在实际问题和实际算法上，我们使用这些输出收集感知判断。实际算法的数据更加有限，因为每个应用程序都有自己独特的属性。例如，不同的彩色化方法 (colorization) 不会显示出太多的结构变化，但会容易产生诸如渗色 (color bleeding) 和颜色变化 (color variation) 等效果。另一方面，超分辨率不会有颜色模糊 (color ambiguity)，但可能会看到不同的算法之间生成不一样的结构。

2 1.4 Superresolution

我们评估了 2017 年 NTIRE 研讨会的结果。我们使用研讨会中的 3 个轨迹 – $\times 2, \times 3, \times 4$ 的上采样率，使用“未知的”下采样率创建输入图像。每个轨道大约有 20 个算法提交。我们还评估了其他几种方法，包括双三次上采样，以及四种表现最好的深度超分辨率方法。

呈现超分辨率结果的一种常见的定性方法是放大特定的片段并比较差异。因此，我们从 Div2K 数据集中随机位置的图像中随机抽取 64×64 个三元组，即 ground truth 高分辨率图像，以及两个算法输出。

2 1.5 Frame interpolation

我们对不同帧插值算法的小片段进行了采样，包括 Davis-Middleburry 数据集上基于流的插值、基于 CNN 的插值和基于相位的插值的三种变体。

由于帧插值产生的 artifacts 可能发生在不同的尺度上，因此我们在采样面片三元组之前随机重新缩放图像。

2 1.6 Video deblurring

我们从视频去模糊数据集、Photoshop Shake Reduction 的去模糊输出、加权傅里叶聚合和深度视频去模糊方法的三种变体中进行采样。

2 1.7 Colorization

我们在彩色化任务中使用随机比例对 ImageNet 数据集的图像进行小片段采样。算法来自 pix2pix、Larsson 等人的算法和 Zhang 等人的算法的变体。

2 2 心理物理相似性测量 (Psychophysical Similarity Measurements)

2 2.1 2AFC similarity judgments

我们随机选择一个图像块 x，然后应用两种失真算法来产生 x_0 和 x_1，然后询问人那个更接近于原来的图像块 x，并记录响应（人的判断值）$h \in \{0, 1\}$。平均而言，人们需要每个小片段花 3 秒钟时间判断，

然后生成一个数据集元组 (x, x_0, x_1, h)。

我们的数据集与先前的数据集之间的对比见下图，我们的人类判断数据集范围更广，量更大，有来自 3000 种失真的 $500k$ 种判断。

Dataset	# Input Imgs/ Patches	Input Type	Num Distort.	Distort. Types	# Levels	# Distort. Imgs/Patches	# Judg- ments	Judgment Type
LIVE [50]	29	images	5	traditional	continuous	.8k	25k	MOS
CSIQ [29]	30	images	6	traditional	5	.8k	25k	MOS
TID2008 [45]	25	images	17	traditional	4	2.7k	250k	MOS
TID2013 [44]	25	images	24	traditional	5	3.0k	500k	MOS
BAPPS (2AFC–Distort)	160.8k	64×64 patch	425	trad + CNN	continuous	321.6k	349.8k	2AFC
BAPPS (2AFC–Real alg)	26.9k	64×64 patch	–	alg outputs	–	53.8k	134.5k	2AFC
BAPPS (JND–Distort)	9.6k	64×64 patch	425	trad. + CNN	continuous	9.6k	28.8k	Same/Not same

我们用小 patch 的意义在于，一整幅图像可能太大，包含太多区域，很难判断，小图上判断就比较容易且合理。其次，通过选择较小的图像块，我们将重点放在较低级别的相似性方面，以减轻可能受高级语义影响的不同“相似性方面”的影响（比如形状结构之类的影响）。最后，用于图像合成的现代方法一般都是使用基于 patch 的损失值来训练深度网络（实现为卷积）。

我们的数据集由 $161k$ 个 patch 组成，这些 patch 来自 MIT Adobe 5k 数据集（5000 张未压缩图像）用于训练，RAIS1k 数据集用于验证。

为了实现大规模收集，我们的数据是在 Amazon Mechanical Turk(AMT) 上广泛收集的，而不是在受控的实验室环境中。Crump 等人表明，尽管无法控制所有环境因素，但 AMT 可以可靠地用于复制许多心理物理学研究。

我们要求在“训练”集合中对每个示例进行 2 次判断，在“值”集合中进行 5 次判断（这样是为了防止“偶然性”带来的错误判断）。更少的判断使我们能够探索更广泛的图像块和失真。

我们添加了由具有明显变形的成对斑块组成的哨兵 (sentinels)，例如，大量高斯噪声与少量高斯噪声。大约 90% 的 Turkers 能够正确通过至少 93% 的哨兵（15 人中的 14 人），这表明他们能够理解任务。

2 2.2 Just noticeable differences(JND)

2AFC 任务的一个潜在缺点是它是“认知可穿透的 (cognitively penetrable)”，即参与者可以有意识地选择在完成任务时选择关注哪些方面的相似性，这将主观性引入到判断中。为了验证这些判断实际上反映了一些客观和有意义的东西，我们还收集了用户对“just noticeable differences”（JND）的判断。我们展示了一张参考图像，然后是一张随机失真的图像，并询问人类这些图像是相同还是不同。

这两个图像块显示时间为 1 秒，其间有 $250ms$ 的间隔。两个看起来相似的图像可能很容易混淆，一个好的感知度量将能够从最容易混淆到最不容易混淆的对进行排序。像这样的 JND 测试可能被认为不那么主观，因为每个判断都有一个正确的答案，并且参与者被认为知道正确的行为需要什么。

我们为传统和基于 CNN 的验证集中的 $4.8k$ 个图像块收集了 3 个 JND 观察结果。每个受试者给出 160 对图像块，以及 40 个哨兵（32 个相同，8 个应用了较大的高斯噪声失真）。我们还提供了一个 10 对的短训练期，其中包含 4 对“相同”的对、1 对明显不同的对和 5 对由我们的失真产生的“不同”的对。我们选择这样做是为了让用户期望大约 40% 的补丁对是相同的。事实上，实验结果中，36.4% 的配对被标记为“相同”（70.4% 的哨兵对和 27.9% 的测试对，注意哨兵中相同的对比较多）。

三 深度特征空间

我们评估不同网络中的特征距离。对于给定的卷积层，我们计算余弦距离（在通道维度 (channel dimension) 上）和网络的空间维度 (spatial dimensions) 和层之间的平均值。我们还讨论了如何根据数据调整现有网络。

网络架构

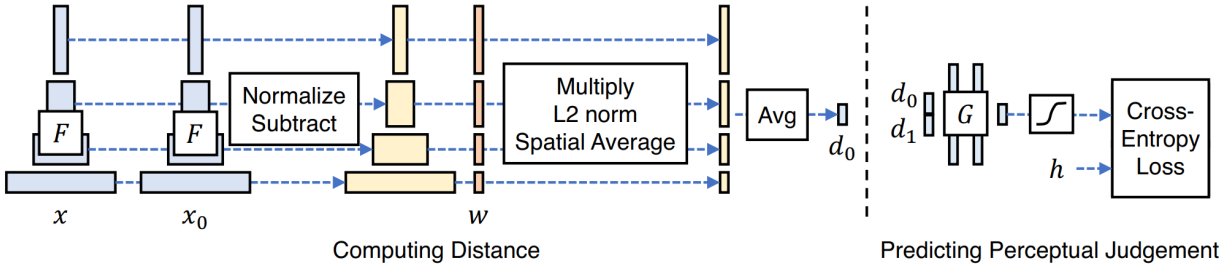
我们调查了 SqueezeNet，AlexNet 和 VGG。从 VGG 中使用了 5 个卷积层，这已经成为了图像生成任务的事实标准了。我们还与较浅的 AlexNet 网络进行了比较，该网络可能与人类视觉皮层的结构更加匹

配。我们使用 [27] 中的 conv1-conv5 层。最后，SqueezeNet 架构被设计为非常轻量级（2.8MB），具有与 AlexNet 类似的分类性能。我们使用第一个 conv 层和随后的“fire”模块。

我们也测试了自监督任务中的网络，比如 puzzle-solving、cross-channel prediction 以及 generative modeling。

网络激活距离

下图左侧和公式 (1) 描述了我们用网络 $\mathcal{F}$ 获得 Reference 和失真图像块之间的距离。



我们抽取 L 层的特征堆栈 (feature stack)，并在通道层上进行归一化。网络第 l 层输出维度是 $\mathbb{R}^{H_l \times W_l \times C_l}$，因此对于第 l 层的网络输出结果我们标记为 $\hat{y}^l, \hat{y}_0^l \in \mathbb{R}^{H_l \times W_l \times C_l}$。

我们在空间上进行平均，并在通道 (channel-wise) 维度（注意这里的通道维度不是前面描述的网络每一层的通道 (channel)，而是一个网络层叫一个 channel-wise）上求和。公式 (1) 可以看到每个网络层都用 $\omega_l \in \mathbb{R}^{C_l}$ 来进行缩放

$$d(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} ||w_l \odot (\hat{y}_{hw}^l - \hat{y}_{0hw}^l)||_2^2 \quad (1)$$

网络激活距离

在上图右边，由于我们有两个图像片段 x_0 和 x_1，因此可以得到两个与原图的距离 d_0 和 d_1。我们通过一个小型网络 $\mathcal{G}$ 来从距离对 d_0, d_1 中预测感知判断 (perceptual judgments)h。

我们使用了一些变体来训练感知判断网络：**lin**, **tune**, 以及 **scratch**。注意这是三种不同的训练方式，这三种训练方式我们都称之为 Learned Perceptual Image Patch Similarity (LPIPS) metric。

对于 **lin**，我们保持之前预训练的网络权重 $\mathcal{F}$ 固定，学习公式 (1) 中的线性权重 ω。

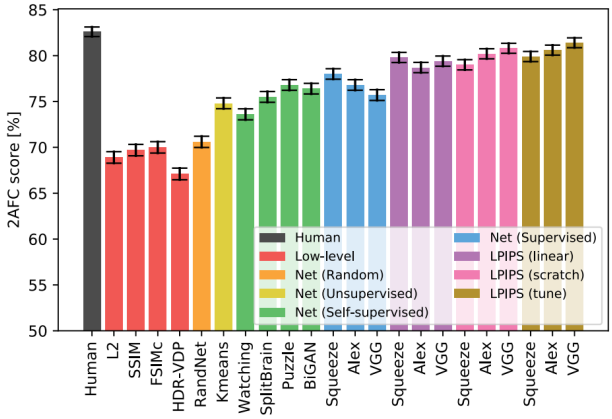
对于 **tune**，我们从预训练分类模型中初始化，允许 $\mathcal{F}$ 中的所有权重都能够被细调。

最后，对于 **scratch**，我们把网络通过随机高斯权重进行初始化，并在我们的整个判断网络流程上来进行训练。

四　实验与验证

实验与验证比较简单，就不再描述了。

不过有一点需要注意一下：在实验与验证环节，发现网络直接学习 metric（LPIPS）的判断准确性比直接比较预训练分类网络的特征相似度有提升。



上图中，蓝色表示的是直接比较预训练分类网络的特征相似度。

源码见 [2]，该代码是 pytorch 构建的，可以通过导入 lpips 直接使用：

```
import lpips
loss_fn_alex = lpips.LPIPS(net='alex') # best forward scores
loss_fn_vgg = lpips.LPIPS(net='vgg') # closer to "traditional" perceptual loss, when used for optimization

import torch
img0 = torch.zeros(1,3,64,64) # image should be RGB, IMPORTANT: normalized to [-1,1]
img1 = torch.zeros(1,3,64,64)
d = loss_fn_alex(img0, img1)
```

源码就暂时不再描述了，有兴趣可以自行阅读。

参考文献

[1] Zhang, Richard, et al. "The unreasonable effectiveness of deep features as a perceptual metric." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[2] https://github.com/richzhang/PerceptualSimilarity