用于超分辨率通用风格迁移的协同蒸馏

Huan Wang[1,2]*, Yijun Li[3], Yuehai Wang[1], Haoji Hu[1†], Ming-Hsuan Yang[4,5]
[1]Zhejiang University [2]Notheastern University [3]Adobe Research [4]UC Merced [5]Google Research
wang.huan@husky.neu.edu yijli@adobe.com {wyuehai,haojihu}@zju.edu.cn mhyang@ucmerced.edu

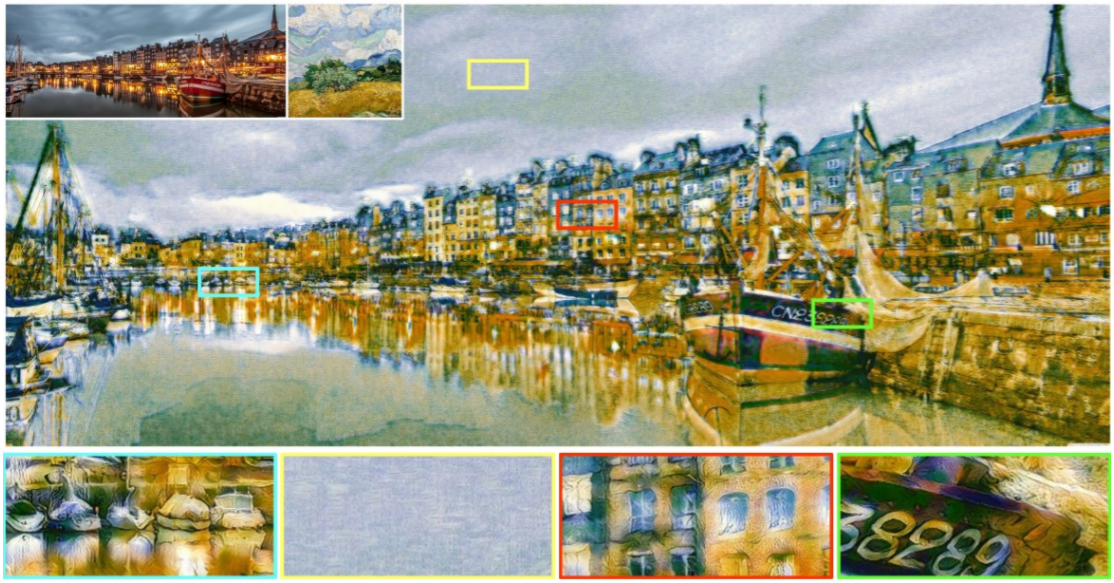


图 1：在单张 Tesla P100（12GB）GPU 上渲染了大约 31 秒的超分辨率风格化样例（10240×4096 像素）。左上角是内容和风格图像。四个特写镜头（539×248）展示在风格化图像下方。

摘要

通用风格迁移方法通常利用来自在大量图像上预先训练的深度卷积神经网络的模型（比如 VGG-19）的丰富的表征。尽管有效，但其在超分辨率图像上的应用受到要处理的模型大小相对有限内存的严重限制。在这个项目中，我们提出了一种新的知识蒸馏方法（命名为协同蒸馏），其用于基于"编码器-解码器"的神经风格迁移，可减少卷积的滤波器。主要思想的基础是，"编码器-解码器"对间有一个独占性的协同关系，这可以被认为是风格迁移模型的一种新的知识。

---

* 这项工作主要是 Huan Wang 在浙江大学的 ISEE 时完成的。†对应的作者。

此外，为了克服应用协同蒸馏时特征大小的不匹配问题，我们引入了一个线性嵌入损失函数，以驱动学生网络去学习教师特征的线性嵌入。后续的实验显示了我们的方法应用于不同的通用风格迁移方式（WCT 和 AdaIN）的有效性，即使模型大小减小了 15.5 倍。特别是在使用压缩模型的 WCT 上，我们首次实现了在 12GB GPU 上进行超分辨率（超过 4000 万像素）通用风格迁移。在基于优化的风格化方案上的进一步实验表明了我们的算法针对不同风格化范式的通用性。可在 https://github.com/mingsun-tse/collaborative-distillation 找到我们的代码和已训练的模型。

1. 引子

通用神经风格迁移（NST）关注于使用来自任何参考图像的风格重构一张内容图像。这通常需要一个具有相当大容量的模型来提取有效的表征以捕获任意风格的特征。最近的基于神经网络的通用风格迁移的方法 [13, 4, 24, 39, 40, 37] 一致表明，使用由预先训练过的深度神经网络（如 VGG-19 [51]）提取的表征同时实现了视觉上令人满意的迁移结果和针对任意风格图像的泛化能力。然而，鉴于硬件算力与内存大小有限，VGG-19 的模型大小极大地限制了输入图像的分辨率。到目前为止，在单张具有 12GB 显存的 GPU 上最高仅处理成功过大约 100 万像素（例如 1024×1024）的图像。虽然有可能利用多个 GPU 实现更高分辨率的风格迁移，但是 VGG-19 模型庞大的基本问题仍然存在，这妨碍了 NST 的实际应用，尤其是在移动设备上的应用。

同时，近年来模型压缩领域发展迅速 [17、21、36、19]，其目的是减少大型 CNN 模型的参数，而又不会造成相当大的性能损失。尽管取得了进步，但大多数模型压缩方法只关注高层次任务，例如分类 [18、58、56、48]和检测 [63, 19]。低级视觉任务的压缩模型仍较少被探讨。知识蒸馏（KD）[2, 1, 21] 是一种很有前途的模型压缩方法，其将大网络（称为教师）的知识转移到小型网络（称为学生）。这里的知识可以是平滑的概率（可以反映被称为*暗知识*的固有的类相似结构）或样本间关系（可以反映不同样本之间的相似结构）[44、46、60、47]。这些知识在单一标签上作为额外信息起作用，因此可以提高学生的表现。但是，这些额外的信息主要是*依赖标签*的，因此很难适用于低级任务。低级视觉任务（例如神经风格迁移）中的暗知识是什么，仍是一个悬而未决的问题。同时，基于“编码器-解码器”的模型被广泛用于神经风格迁移，其中解码器通常是通过编码器的知识进行训练的。值得注意的是，他们在风格化过程中共同形成了一种*独占的协同关系*，如图.2 所示。由于解码器 D 被训练来专门与编码器 E 工作，如果另一个编码器 E’ 也能与 D 工作，就意味着 E’ 可以代替 E。基于这个想法，我们提出了一种新的知识来提炼神经风格迁移的深层模型：编码器与解码器间的协同关系。

图 2：两个不同编码器-解码器协同关系中独占协同现象的示例：WCT 的图像重建 [39]（第一行）和 AdaIN[24]（第二行）的风格迁移。第一列是输入，其他四列显示使用不同的编码器-解码器组合的输出结果。如果两个“编码器-解码器”对（E1-D1、E2-D2）是独立训练的，则编码器只能与其*匹配*的解码器配套工作。

在已有一个冗余的大型编码器（例如，VGG-19）的情况下，我们提出了一个两步压缩方案：首先，为编码器训练一个协同网络，在这个语境下即解码器；然后，用一个小型的编码器替换大型编码器，保持解码器不变对小型编码器进行训练。由于小型编码器的通道通常较少，因此其输出特征的维数比大型编码器的少。因此，这个小网络不能直接与解码器协同工作。为了解决这个问题，我们要求学生学习教师输出的线性嵌入，这样教师的输出可以通过学生输出的简单线性组合进行重建，然后再被送给解码器。

值得注意的是，在我们的方法中不限制具体的协作形式。本文将展示它可应用于两种不同的最先进的风格迁移方案：WCT [39]（其中的协作是图像重建）和 AdaIN [24]（其中的协作是风格迁移）。这项工作的主要贡献是：

为通用神经风格迁移提出了一种的新的知识蒸馏方法。编码器与解码器之间的独占协作关系被认为是一种新的知识，可以应用于不同的协作关系中。

为了解决算法中学生和教师网络之间的特征维度不匹配问题，我们提出要求学生学习教师输出的线性嵌入，这同时也是一个规范化器，它可以把更多的监督融入学生的中间层以提升学习能力。

广泛的实验显示了我们的方法的优点：在不同的风格化框架（WCT [39]， AdaIN [24]，和 Gatys [13]），中实现了 15.5 倍的参数缩减与甚至更好的视觉效果。特别是在 WCT 上，压缩模型使我们能够在单个 12GB GPU 上首次进行超分辨率（4000 万像素）通用风格迁移。

2. 相关研究

风格迁移。在深度学习时代之前，图像风格迁移主要通过非参数采样 [9]、非真实呈现 [15, 52] 或图像类比 [20] 来解决。但是，这些方法针对某些特定风格而设计，并且依赖于低级统计信息。最近，Gatys 等人[13]提出了神经风格迁移，它采用了预先训练的 VGG-19 模型[51]的深层特征，通过匹配生成的图像和给定风格图像之间的二阶信息来实现风格化。已经有许多方法被开发出来用于提高视觉质量[34, 57, 49, 64]，处理速度[35, 54, 29, 11, 37]，用户控制[41, 59, 14]与风格多样性[8, 3, 38, 24, 39]。然而，所有这些基于神经网络的方法的一个常见限制是，它们不能在有限内存下处理超分辨率的内容和风格图像。某些方法 [54、29、49] 通过学习特定风格示例或类别的前馈网络来实现高分辨率风格化（高达 1000 万像素，例如 3000× 3000 像素），但它们并不通用于其他未知风格。相比之下，我们的目标是实现仅用一个模型的通用风格的超分辨率图像风格迁移。

模型压缩。模型压缩和加速最近也引起了很多关注，其目的是在性能没有巨大妥协的前提下获得更小和更快的模型。现有方法大致分为五类，即低秩分解[7, 27, 32, 63]，修剪[33, 18, 17、36 、58 、19 、56 、55]、量化 [5, 48, 65, 25]，知识蒸馏 [2, 1, 21, 61] 和紧缩结构重构或搜索 [26, 23, 50, 62, 45, 53, 10]。然而，这些方法主要关注高级视觉任务，通常是分类和检测。很少有方法关注低级视觉任务，如风格迁移，其中许多方法同样受限于 CNN 的巨大模型。与面向高级视觉的 CNN 压缩（仅需要维护特征的全局语义信息来保持准确性）不同，低层视觉的模型压缩带来了额外的挑战，例如如何在风格迁移中维持像纹理、颜色种类这样的局部结构。

在我们的工作中，我们发明了一个深度监督的知识蒸馏方法来从预训练的冗余的 VGG-19 [51] 中学习一个小得多的模型。压缩模型实现了超过 15 倍的参数和计算的精简。更重要的是，模型大小的减小使超高分辨率图像的通用风格迁移成为可能。据我们所知，最近只有一个项目[30]使用 GAN[16]来学习超分辨率图像上的无配对风格迁移网络。但是，它们通过处理图像子样本然后将它们合并回整个图像来实现此目的。相比之下，我们的方法从根本上减少了模型的复杂度，可以直接处理整张图像。

3. 我们提出的方法

3.1 协同蒸馏

风格未知的风格化方法通常采用“编码器-解码器”方案来学习风格渲染的深度表征，然后将它们反转回风格化图像。由于风格信息不是直接在模型中编码的，编码器部分需要具有足够的表达能力，才能提取通用风格的信息表征。现有方法 [13, 4, 24, 39] 通常选择 VGG-19 [51] 作为编码器，这是考虑到它巨大的容量和分层结构。至于解码器，根据不同的风格化方案，它可以与编码器具有不同的协作关系。我们在这里讨论两个最先进的通用风格迁移方案：WCT[39]和 AdaIN[24]。(i) 对于 WCT，风格化过程是使用风格特征的二阶信息将白化和着色变换[22]应用于内容图像的特征。然后，转换后的的内容图像特征由解码器反转为图像。因此，解码器训练不直接涉及风格化。WCT 中的协作关系本质上是图像重建。(ii) 对于 AdaIN，与 WCT 不同，其解码器训练

直接参与了风格化。两个图像（内容和风格）都被送进编码器，然后进入特征空间，内容特征由风格特征的统计信息（均值和方差）渲染。最后，解码器将渲染的内容特征反转回风格化的图像。风格化图像在内容（或风格）上应接近内容图像（或风格图像）。因此，AdaIN 的协作关系是风格的迁移。

尽管上述两种方案的范式差异较大，但它们都是基于“编码器-解码器”的，解码器也都是通过编码器的知识进行训练的。这意味着，在解码器的训练过程中，编码器的知识泄漏给了解码器。根据推测与经验，解码器 D 只与它匹配的编码器 E 协作，就像螺母与螺栓一样。对于另一个编码器 E‘，即使它具有与 E 相同的结构，D 和 E’ 也不能协同工作（参见图 2）。这种排他性显示了解码器包含有一些与编码器有关的信息。如果我们能找到一种方法，使网络 E‘ 与 D 兼容，这意味着 E’ 可以在功能上取代原来的编码器 E。如果 E‘ 同时比 E 小得多，那么就实现了模型压缩。基于这一想法，我们提出了一种针对 NST 的新型蒸馏方法，称为"协同蒸馏"，由两个步骤组成。

第一步，根据手头的任务，我们为大型编码器训练协作者网络。如图 3（a）所示，对于 WCT [39]，解码器经过训练，可反转特征以尽可能忠实于输入图像（即图像重建），其中同时利用了像素重建损失和感知损失 [29]。

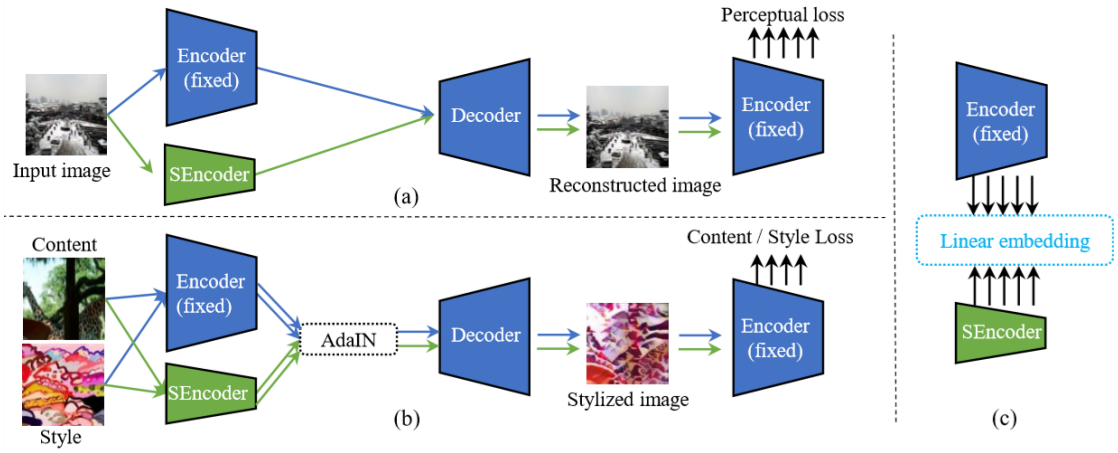


图 3：我们提出的"协同蒸馏"的框架图解（最好能看彩色版本的图）。（a）和（b）展示了两种通用神经风格迁移的“编码器-解码器”协作关系：WCT [39] 的图像重建和 AdaIN [24] 的风格迁移。蓝色箭头显示训练协作者网络（即解码器）时的前向路径。绿色箭头显示小编码器（"SEncoder"）被训练以替换原始编码器（"Encoder"）时的前向路径。（c）显示我们提出的线性嵌入方案，它用于解决特征大小不匹配问题，并在小型编码器的中间层融入更多的监督。

$$\mathcal{L}_r^{(k)} = \|\mathcal{I}_r - \mathcal{I}_o\|_2^2 + \lambda_p \sum_{i=1}^{k} \|\mathcal{F}_r^{(i)} - \mathcal{F}_o^{(i)}\|_2^2, \quad (1)$$

$$\mathcal{L}_{\text{st}} = \|\mathcal{F}_{\text{st}}^{(4)} - \mathcal{F}_c^{(4)}\|_2^2 + \lambda_s \sum_{i=1}^{4} \|\mathcal{G}_{\text{st}}^{(i)} - \mathcal{G}_s^{(i)}\|_2^2, \quad (2)$$

其中 k∈[1，2，3，4，5] 表示 VGG-19 的第 k 阶段；$F^{(i)}$ 表示 ReLU_i_1 层的特征表；λ_p 是用于平衡感知损失和像素重建损失的权重；$\mathcal{I}_0$ 和 $\mathcal{I}_r$ 分别表示原始图像和重建图像。对于 AdaIN [24]，风格迁移直接涉及了解码器。因此，解码器损失由内容损失和风格损失构成。

其中 $\mathcal{G}$ 是描述风格 [12，13] 的 Gram 矩阵，λ_s 是用于平衡风格损失和内容损失的权重；下标"st"，"c"和"s"分别表示风格化图像、内容图像和风格图像。

在有了协作者之后，我们算法的第二步是用一个小编码器 E‘ 替换原始编码器 E。为简单起见，在这项工作中，我们让 E‘ 与 E 有相同的结构，但每层中的滤波器更少。我

们期望小型编码器 E’ 在功能上可以与原始编码器 E 等效，如图 3 的（a）和（b） 所示。与第一步类似，协作者网络损失函数（表示为 $\mathcal{L}_{\text{collab}}$）可以根据特定的协作任务选取不同的形式。在这里，对于 WCT，$\mathcal{L}_{\text{collab}} = \mathcal{L}_r$ 而对于 AdaIN，$\mathcal{L}_{\text{collab}} = \mathcal{L}_{st}$。

3.2 线性嵌入

在我们的协同蒸馏方法中，小编码器与解码器网络连接。在他们的交接处中，存在一个特征大小不匹配的问题。具体来说，如果原始的编码器输出一个大小为 C×H×W 的特征，那么解码器的输入也应该为 C×H×W。但是，由于小型编码器的滤波器较少，因此它将输出大小为 C‘×H ×W （C’ < C）的特征，而解码器无法接受此输入。要解决这个问题，我们先要看通道数在风格化过程中起着什么作用。正如 Gatys [12， 13] 的解释，图像的风格由从 VGG-19 提取的深度特征的 Gram 矩阵描述。

$$\mathcal{G} = \mathcal{F} \cdot \mathcal{F}^T, \tag{3}$$

其中 F 是从 VGG-19 的某些卷积层中提取的大小为 C×HW 的深度特征；$\mathcal{G}$ 表示大小为 C×C 的 Gram 矩阵；T 代表矩阵转置。由于我们是要压缩这些特征，即消除它们的冗余，可以推导出 F 是一些特征基础向量在较低维度的线性组合。

$$\mathcal{F} = Q \cdot \mathcal{F}', \tag{4}$$

其中 Q 是一个大小为 C×C‘ 的变换矩阵，F’ 是大小为 C‘×HW 的特征基础矩阵，它可以被看作是原始深层特征 F 的线性嵌入。然后很容易发现，Gram 矩阵 $\mathcal{G}'$ $= \mathcal{F}' \cdot \mathcal{F}'^T$ ，因为新特征 F‘ 有与原始 Gram 矩阵相同数量的特征值。换句话说，如果我们用 F‘ 代替原来的冗余的 F，风格描述能力得以保持。变换矩阵 Q 是通过一个没有非线性激活函数的完全连接层学习的，以满足线性假设。因此，线性嵌入损失可以以下式表示：

$$\mathcal{L}_{\text{embed}} = ||\mathcal{F} - Q \cdot \mathcal{F}'||_2^2. \tag{5}$$

更进一步地，上述提出的解决方案不仅限于最终输出层。它也可应用于小型编码器的中间层。具体地说，我们将线性嵌入应用于原始编码器和小型编码器之间的其他四个中间层（ReLU_k_1、k ∈ [1，2，3，4]），如图 3（c）所示。我们有两个这样做的动机。首先，在我们的方法中，当使用 SGD 算法训练小型编码器时，其唯一的梯度源将是解码器，通过完全连接层 Q 传递。但是，Q 的参数并不多，所以它实际上会形成一个信息瓶颈，减慢学生学习的速度。当这些分支被插入到网络的第三层，它们将给学生注入更多的梯度，从而促进其学习，这对于容易发生梯度消失的深层网络尤其有效。其次，在神经风格迁移中，图像的风格通常由许多中间层的特征描述 [13、24、39]。因此，为了保证它们不会失去太多的风格描述能力，以便以后在风格迁移中使用，有必要向这些层增加更多监督。

最后，我们提出的算法中训练小编码器的总损失可以概括为：

$$\mathcal{L}_{\text{total}} = \beta \sum_{i=1}^{k} \mathcal{L}_{\text{embed}} + \mathcal{L}_{\text{collab}}, \tag{6}$$

其中 β 是用于平衡两种损失的权重因子。

4. 实验结果

在本节中，我们将首先演示与通用风格迁移框架 WCT [39] 中的原始 VGG-19 相比，压缩的 VGG-19 的有效性。然后，我们将在 AdaIN 上演示，协同蒸馏并不局限于某一种协作关系。最后，为了显示该方法的通用性，我们还将使用基于优化的风格迁移方法 Gatys[13]，其中的协作关系

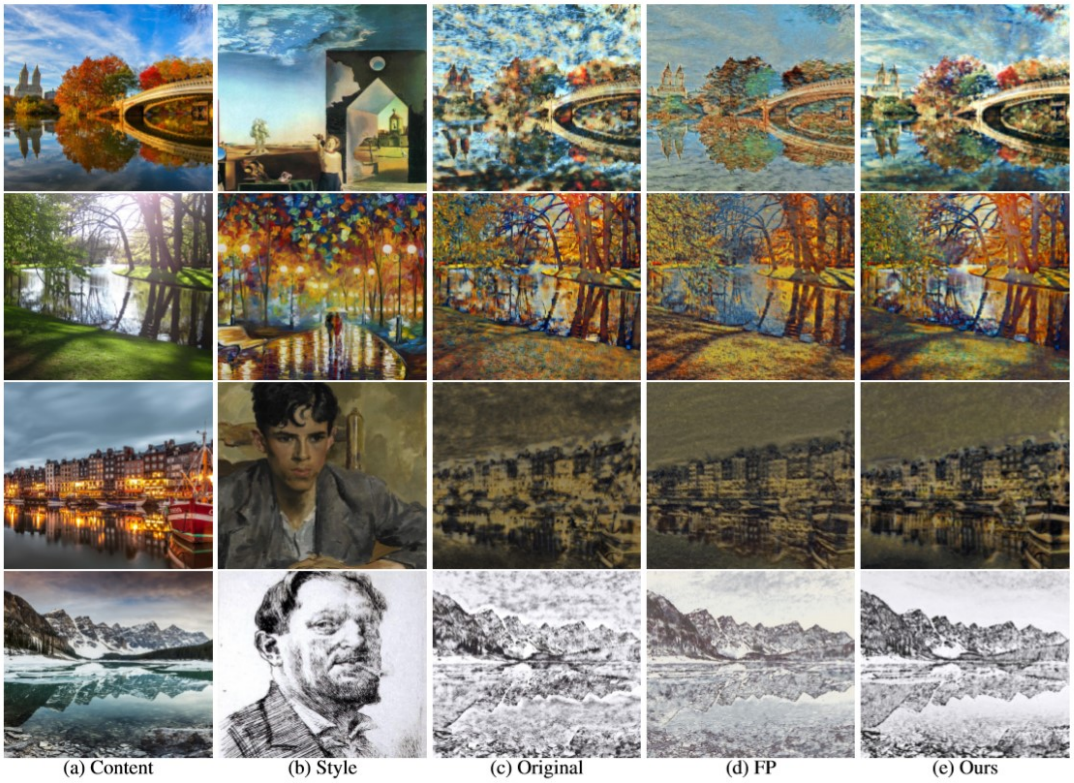


图 4：由三种不同模型（即原始 VGG-19、FP 简化的 VGG-19 和我们的压缩 VGG-19）生成 3000×3000 风格化图像的比较（最好在带颜色版本上放大查看）。

与 WCT 一致，即图像重建。我们首先在原始 VGG-19 能处理的最大图像分辨率（3000×3000）上进行比较，然后展示小模型在更大分辨率（即超分辨率）上的一些风格化的示例。所有实验都在一张 Tesla P100 12GB GPU 上进行，即限定相同的有限内存。

评估压缩方法。由于为低级图像合成任务专门设计的模型压缩方法很少，因此我们将该方法与高级图像分类任务中的典型压缩算法进行比较，即滤波剪枝（FP）[36]。具体来说，我们首先在分类中将 FP 应用于 VGG-19，以获得与我们有相同的结构的压缩模型。然后在 ImageNet [6] 上进行微调，以恢复性能。最后，通过优化（1）中的损失获得解码器。

4.1 WCT 上的风格迁移

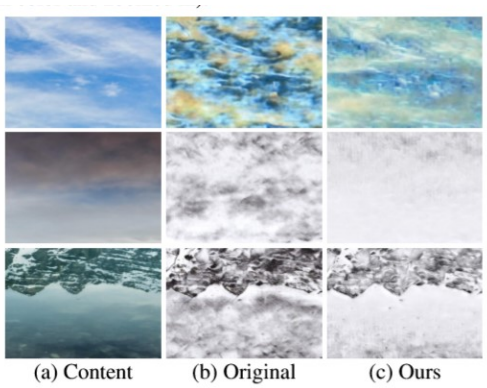
由于我们需要原始解码器作为协作者，我们首先使用 VGG-19 的镜像结构在 MS-COCO 数据集 [42] 上为图像重建训练一个解码器。在训练期间，编码器是固定的，有 $\lambda_p = 1$。我们随机裁剪 300×300 图像的 256×256 补丁作为输入。Adam [31] 用作具有固定学习速率 10^{-4}、批大小 16 的优化求解器。在 WCT [39] 中，采用级联、由粗到细的风格化过程以获得最佳效果，因此 5 级 VGG-19（最大 ReLU_k_1、k ∈ [1，2，3，4，5]）的解码器都经过训练。然后构建一个带损失的“编码器-解码器”网络（6），其中 β 设置为 10。压缩编码器可以随机初始化，但我们从经验上发现，使用原始 VGG-19 中的最大滤波器（基于 L_1 规范）进行初始化将有助于压缩模型收敛得更快，因此我们在所有实验中都使用此初始化方案。经过 20 分钟的训练，我们得到压缩编码器。他们的镜像解码器一开始都使用损失函数（1），依照相同的规则训练。

图 4 显示风格化结果的比较。总的来说，我们的模型用更少的参数实现了相似的甚至更好的风格化图像。由原始 VGG-19 模型和我们的压缩模型生成的风格化图像比那些由 FP 模型生成的风格化图像看起来更好（颜色更丰富，图像更锐利）。与原始 VGG-19 模型相比，我们的压缩模型生成的结果往往有更少的凌乱纹理，而原始模型通常会在风格化图像中突出显示太多纹理。例如，天空和水在图像中通常看起来光滑，但实际上，自然图像中没有绝对平滑的部分，因此天空和水仍有细微差别。在图 5 中，原始的 VGG-19 模型倾向于突出这些细微差别，以至于整个图像看起来凌乱，而我们的模型只强调语义上最突出的部分。这种现象可以解释为，参数较少的模型的容量有限，因此不太容易过度拟合。自然，一个过度参数化的模型将花费额外的容量过拟合数据中的噪音，例如，这里的天空和水的细微差别纹理。

同时，即使压缩模型往往不大会过拟合，由 FP [36] 修剪的模型还有一个描述能力丢失的问题，这体现在图 4 的两个方面。首先，正如我们所看到的，FP 简化模型倾向于以较少的颜色生成风格化的图像。其次，仔细观察时，FP 简化模型的风格化图像具有严重的棋盘伪影。

用户研究。风格迁移有一个公开的问题，缺乏广泛接受的比较标准[28]，主要是因为风格化相当主观。为了得到更客观的比较，以前的一些项目 [39] 进行了用户研究，以调查用户对不同风格化结果的偏好。在这里，我们采用这个想法，调查由三个模型产生的风格化图像哪个视觉上更赏心悦目。我们使用三种模型生成 20 对图像。在其中，随机选择 10 对给每个调查对象，让他们选择哪一个是最好的。在这一部分，我们计得 600 个有效投票。用户研究结果如表.1 所示，其中我们的压缩模型的风格化图像平均评价最高，与图 4 中的定性比较结果一样。

风格距离。为了进一步定量地评价这三种模型，我们研究了风格化图像与风格图像之间的风格相似性。直观地看，更好的风格化图像应在风格空间上更接近风格图像，因此我们将风格距离定义为：



(a) Content (b) Original (c) Ours

图 5：使用原始模型和压缩模型的内容图像中细微纹理（例如天空和水区）的风格细节比较。

$$\mathcal{D}_{\text{style}}^{(k)} = \|\mathcal{G}(\mathcal{F}^{(k)}(\mathcal{I}_{\text{stylized}})) - \mathcal{G}(\mathcal{F}^{(k)}(\mathcal{I}_{\text{style}}))\|_2, \quad (7)$$

其中 $\mathcal{G}$ 是 Gram 矩阵。特别地，我们将用户研究中的 20 对风格化图像馈送至原始 VGG-19 以提取 5 阶段层（ReLU_k_1，k ∈ [1，2，3，4，5]）的特征，然后根据这些特征计算风格距离。表. 2 显示，原始模型和我们的压缩模型生成的风格化图像与 FP 精简模型相比明显更接近风格图像。我们的模型与原始模型基本相当，这与用户偏好（表.1）相符。

表 1：用户在三种通用艺术风格迁移框架中对不同深度模型的偏好研究

Stylization scheme	Original	FP-slimmed	Ours
WCT [39]	33.0%	24.3%	**42.7%**
AdaIN [24]	**51.1%**	6.5%	42.4%
Gatys [13]	**46.5%**	22.3%	31.2%

表 2：三种模型的风格距离比较。值是基于 5 阶段深度特征而定义为公式 (7) 的风格距离（较小越好）

Model	Conv1	Conv2	Conv3	Conv4	Conv5
Original	**25.5**	44.6	**236.3**	465.5	476.5
FP-slimmed	43.8	64.3	416.4	597.8	482.1
Ours	28.1	**43.8**	269.7	**446.4**	**399.0**

除了上面显示的高分辨率风格化图像外，我们在图 1 展示了一张超分辨率图像。即使放大了看，形状和纹理仍然非常清晰。据我们所知，这是第一次，我们可以使用单张 12GB GPU 获得超分辨率通用风格迁移图像。我们在 表.3 中展示了有关模型大小和加速的统计数据。我们的压缩模型比原来的 VGG-19 要小 15.5 倍，GPU 运行速度快 3.6 倍。注意，5 阶段的整个模型大小只有 10 MB，很容易放进移动设备。

分离研究。现在探讨两个我们提出的子方案，协作蒸馏（$\mathcal{L}_{\text{collab}}$）和线性嵌入（$\mathcal{L}_{\text{embed}}$）的效果。图 6 的分离结果表明，线性嵌入和协作的损失函数都可以将大量知识从教师转移到学生之间。通常，即使比原模型小 15.5 倍，这两种方案也可以独立生成基本相当的结果。在两种损失之间，$\mathcal{L}_{\text{embed}}$ 只迁移特征域的信息而不处理图像域的信息，因此其结果具有更多的失真和一些棋盘效果（请放大查看图 6（c）的细节）。而 $\mathcal{L}_{\text{collab}}$ 主要关注图像重建，在失真更小的情况下产生更锐利的结果，这与我们的直觉相符，解码器有足够多的知识，可以用来训练一个小编码器。当同时利用两个损失函数时，我们得到两者之间的平衡点：结果既没有棋盘，也适当地控制了艺术失真。

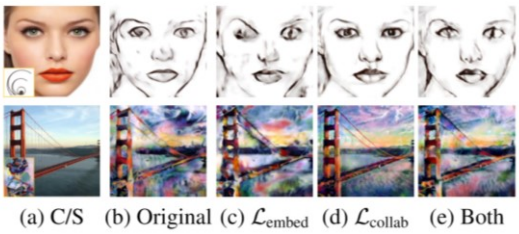


图 6：两个损失函数在 WCT 上的分离研究。(a) 内容和风格图像 (b) 使用原始模型 (c) 使用 $\mathcal{L}_{\text{embed}}$ (d) 使用 $\mathcal{L}_{\text{collab}}$ (e) 同时使用两种损失函数

图 7：AdaIN 上三种模型的比较

4.2 AdaIN 上的风格迁移

我们在 AdaIN 上进一步评估我们提出的方法，这里的“编码器-解码器”协作任务是风格迁移。训练过程和网络结构按与章节.4.1 相同的方式设置，区别在于 $\mathcal{L}_{\text{st}}$（公式.2）是现在的协作者损失。λ_s和 β 都设置为 10。结果如图 7 所示，我们的压缩编码器在视觉上与原始编码器的结果相当，而 FP 精简模型则显著降低了视觉效果。在表.1 中的用户研究中进一步证明了此视觉评估的合理性，其中我们和原始模型获得了相似的投票，且明显比 FP 算法压缩的模型 [36] 的票数要多。

4.3 基于优化的风格迁移

尽管项目 [13] 不是基于“编码器-解码器”的，但值得检查一下按我们的方法压缩的小型编码器在这种情况下是否仍然性能良好。选用层 Conv_k_1（k ∈ [1，2，3，4，5]）作风格匹配，选用层 Conv4_2 作内容匹配。选择 L-BFGS [43] 作优化求解器。图.8 显示，压缩模型与原始 VGG-19 生成的结果基本相当。与 4.1 章节中艺术风格迁移的实验类似，我们还进行了用户研究，以便进行更客观的比较。表.1 的结果表明，我们的压缩模型比原始模型稍微不受欢迎一点（合理推测是因为压缩模型少了 15.5 倍的参数），但表现仍比 FP 方法压缩的模型更好。

图 8：使用 Gatys 等人的方法 [13] 时使用原始模型和压缩模型的风格化图像比较（最好在带颜色的版本上放大查看）

表 3：原始模型和压缩模型的摘要。存储大小按 PyTorch 的模型进行测量。测量 GFLOPs 和推理时间时内容和风格都是 3000×3000 像素的 RGB 图像。使用 PyTorch 实现在 GPU 上测量最大输入分辨率，其中内容和风格都是大小相同的方形 RGB 图像。

Model	# Params (10^6)	Storage (MB)	# GFLOPs	Inference time (GPU/CPU, s)	Max resolution
Original	17.1	66.6	6961.7	31.2/937.7	3000×3000
Ours	1.1(15.5×)	5.2(12.8×)	451.6(15.4×)	8.6(3.6×)/366.0(2.6×)	6000×6000

5. 讨论

我们简要解释了为什么现有的知识蒸馏方法[44、46、60、47]通常对风格的迁移不那么有效。我们论文中的 FP 修剪 VGG-19(称为 A)在 ImageNet 上实现了 83.47% 的前 5 精度。在 A 的微调过程中，我们获得了一个中间模型（称为 B），其前 5 精度较低，为 80.55%。我们将 A 和 B 的风格化质量与 WCT 进行比较，发现 A 的结果与 B 的结果相比没有任何优势，因为令人不快的杂乱纹理仍然存在（图 9，第 1 行）。这意味着分类中的一个小的精度增益（通常小于 3%，这是当前最好的蒸馏方法所能实现的最大增益 [44，46，60，47]）不能真正转换成神经风格迁移的感知提升。此外，我们只在蒸馏编码器时应用这种方法，而不是解码器，因为它会降低视觉质量，如图 9（第 2 行）所示。原因是，解码器负责图像重建，已经经过适当的监督，即公式 (1) 中的像素和感知损失。当在小解码器上应用蒸馏时，原始解码器上的额外监督不仅不会有助于反而会损害损失函数的作用（1），从而恶化了风格化结果的视觉质量。

图 9：第 1 行：FP 修剪模型 A 和 B 的风格化比较；第 2 行：WCT 框架下使用和不使用 KD 的编码器之间的风格化比较。

6. 结论

鉴于 CNN 巨大的模型体积，输入分辨率是通用神经风格迁移的一个重要限制。在本文中，我们提出了一种新的知识蒸馏方法（即协同蒸馏），以减少 VGG-19 的模型大小，它利用了在通用风格迁移中“编码器-解码器”对形成独占协作关系的现象。为了解决特征大小不匹配的问题，我们进一步提出了线性嵌入方案。进一步的实验揭示了我们的方法在两种通用的风格化方法（WCT 和 AdaIN）上的优点。在 Gatys 风格化框架中的更进一步实验证明了我们方法在基于优化的风格迁移范式上的通用性。虽然我们主要专注于神经风格的迁移，但“编码器-解码器”方案也可应用于其他的一般低级视觉任务，如分辨率提升和图像修复。我们的方法在这些任务上的表现值得研究，这也将是我们未来的工作。

致谢

我们感谢 Wei Gao 和 Lixin Liu 的有益的讨论。这项研究得到了中国国家重点攻关项目（No 2017YFB1002400）和美国国家科学基金会 CAREER（No.1149783）的支持。

参考文献

[1] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In NeurIPS, 2014. 2, 3
[2] Cristian Bucilu, Rich Caruana, and Alexandru NiculescuMizil. Model compression. In SIGKDD, 2006. 2, 3
[3] Dongdong Chen,Lu Yuan,Jing Liao,Nenghai Yu,and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In CVPR, 2017. 3
[4] Tian Qi Chen and Mark Schmidt. Fast patch-based style transfer of arbitrary style. arXiv preprint arXiv:1612.04337, 2016. 2, 3
[5] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. arXiv preprint arXiv:1602.02830, 2016. 3
[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009. 5
[7] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In NeurIPS, 2014. 3
[8] Vincent Dumoulin,Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In ICLR, 2017. 3
[9] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In ICCV, 1999. 3
[10] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. JMLR, 20(55):1–21, 2019. 3
[11] WeiGao,YijunLi,YihangYin,andMing-HsuanYang. Fast video multi-style transfer. In WACV, 2020. 3
[12] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In NeurIPS, 2015. 4
[13] Leon A Gatys, Alexander SEcker,and Matthias Bethge. Image style transfer using convolutional neural networks. In CVPR, 2016. 2, 3, 4, 5, 7, 8
[14] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In CVPR, 2017. 3
[15] Bruce Gooch and Amy Gooch. Non-photorealistic rendering. AK Peters/CRC Press, 2001. 3
[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu,DavidWarde-Farley,SherjilOzair,AaronCourville,and Yoshua Bengio. Generative adversarial nets. In NeurIPS, 2014. 3
[17] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In ICLR, 2016. 2, 3
[18] Song Han,Jeff Pool,John Tran,and William JDally. Learning both weights and connections for efficient neural network. In NeurIPS, 2015. 2, 3
[19] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In ICCV, 2017. 2, 3
[20] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In SIGGRAPH, 2001. 3
[21] Geoffrey Hinton, OriolVinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015. 2, 3
[22] MilihaHossain. Whitening and coloring transforms for multivariate gaussian random variables. Project Rhea, 2016. 3 [23] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017. 3
[24] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In ICCV, 2017. 2, 3, 4, 5, 7
[25] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran ElYaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. JMLR, 18(1):6869–6898, 2017. 3
[26] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size. arXiv preprint arXiv:1602.07360, 2016. 3
[27] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. In BMVC, 2014. 3
[28] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, and Mingli Song. Neural style transfer: A review. arXiv preprint arXiv:1705.04058, 2017. 6
[29] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In ECCV, 2016. 3, 4
[30] Andrej Junginger, Markus Hanselmann, Thilo Strauss, Sebastian Boblest, Jens Buchner, and Holger Ulmer. Unpaired high-resolution and scalable style transfer using generative adversarial networks. arXiv preprint arXiv:1810.05724, 2018. 3
[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2015. 5
[32] Vadim Lebedev,Yaroslav Ganin,Maksim Rakhuba,Ivan Oseledets, and Victor Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. arXiv preprint arXiv:1412.6553, 2014. 3
[33] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In NeurIPS, 1990. 3
[34] Chuan Li and Michael Wand. Combining markov random fieldsand convolutional neuralnetworks for image synthesis. In CVPR, 2016. 3
[35] ChuanLiandMichaelWand. Precomputedreal-timetexture synthesis with markovian generative adversarial networks.In ECCV, 2016. 3
[36] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In ICLR, 2017. 2, 3, 5, 6, 7
[37] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast arbitrary style transfer. In CVPR, 2019. 2, 3
[38] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Diversified

texture synthesis with feed-forward networks. In CVPR, 2017. 3
[39] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In NeurIPS, 2017. 2, 3, 4, 5, 6, 7
[40] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In ECCV, 2018. 2
[41] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. ACM Transactions on Graphics, 36(4):1–15, 2017. 3
[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, PietroPerona,Deva Ramanan,PiotrDoll´ar,and CLawrence Zitnick. Microsoft COCO: Common objects in context. In ECCV, 2014. 5
[43] DongCLiuandJorgeNocedal. On the limited memory bfgs method for large scale optimization. Mathematical Programming, 45(1-3):503–528, 1989. 8
[44] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In CVPR, 2019. 2, 8
[45] Ningning Ma,Xiangyu Zhang,Hai-Tao Zheng,and JianSun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In ECCV, 2018. 3
[46] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In CVPR, 2019. 2, 8
[47] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In ICCV, 2019. 2, 8
[48] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In ECCV, 2016. 2, 3
[49] Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Bj¨orn Ommer. A style-aware content loss for real-time hd style transfer. In ECCV, 2018. 3
[50] Mark Sandler,Andrew Howard,Menglong Zhu,Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In CVPR, 2018. 3
[51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. 2, 3
[52] Thomas Strothotte and Stefan Schlechtweg. Nonphotorealistic computer graphics: modeling, rendering, and animation. Morgan Kaufmann, 2002. 3
[53] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. InICML,2019. 3
[54] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In ICML, 2016. 3
[55] Huan Wang, Xinyi Hu, Qiming Zhang, Yuehai Wang, Lu Yu, and Haoji Hu. Structured pruning for efficient convolutional neural networks via incremental regularization. JSTSP, 2019. 3
[56] Huan Wang, Qiming Zhang, Yuehai Wang, and Haoji Hu. Structured probabilism ticpruning for convolutional neuralnetwork acceleration. In BMVC, 2018. 2, 3
[57] Xin Wang, Geoffrey Oxholm, Da Zhang, and Yuan-Fang Wang. Multimodal transfer: A hierarchical deep convolutional neural network fo rfast artistic style transfer. InCVPR, 2017. 3
[58] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and HaiLi. Learning structured sparsity in deep neural networks. In NeurIPS, 2016. 2, 3
[59] PierreWilmot, EricRisser, and Connelly Barnes. Stableand controllable neural texture synthesis and style transfer using histogram losses. arXiv preprint arXiv:1701.08893, 2017. 3
[60] Lu Yu, Vacit Oguz Yazici, Xialei Liu, Joost van de Weijer, Yongmei Cheng, and Arnau Ramisa. Learning metrics from teachers: Compact networks for image embedding. In CVPR, 2019. 2, 8
[61] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In ICLR, 2017. 3
[62] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In CVPR, 2018. 3
[63] X. Zhang, J. Zou, K. He, and J. Sun. Accelerating very deep convolutional networks for classification and detection. TPAMI, 38(10):1943–1955, 2016. 2, 3
[64] Yulun Zhang, Chen Fang, Yilin Wang, Zhaowen Wang, Zhe Lin, Yun Fu, and Jimei Yang. Multimodal style transfer via graph cuts. In ICCV, 2019. 3
[65] Shuchang Zhou,Yuxin Wu,Zekun Ni,Xinyu Zhou,He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160, 2016. 3