

标准化卷积神经网络

Dongsuk Kim¹ Geonhee Lee¹
Myungjae Lee² Shin Uk Kang² Dongmin Kim²

² JLK Group(Research)

¹kds900910@gmail.com

Abstract

我们引入了一个归一化卷积神经网络层，这是一种在卷积网络中进行归一化的新型方法。与传统方法不同，该层在卷积过程中对 `im2col` 矩阵的行进行归一化处理，使其能够自适应切片输入，并更好地与核结构保持一致。这种独特的方法区别于标准归一化技术，且无法直接整合到优化了传统卷积操作的现有深度学习框架中。我们的方法具有普适性，适用于涉及卷积层的所有深度学习任务。通过在卷积过程中内在地进行归一化处理，它作为自归一化网络的卷积适应版本，无需额外的归一化层即可维持其核心原则。值得注意的是，在微批次训练场景中，它始终优于其他不依赖批次的归一化方法。这种性能提升源于对 `im2col` 矩阵行的标准归一化，理论上可以导致更平滑的损失梯度和改进的训练稳定性。

1 介绍

规范化方法显著提升了深度神经网络的性能。其中，批量归一化 (BN) [7] 由于其加速训练和帮助模型收敛到损失函数 [18] 更好最小值的能力，在深度学习架构中被广泛采用。然而，BN 在大规模批处理环境中特别有效，因为它设计用于缓解内部协变量偏移。为了解决 BN 在小批量尺寸下的局限性，已经提出了各种与批次无关的规范化方法和微型批网络。尽管如此，这些方法仍然难以匹配 BN 在大批量训练中应用时的表现。本文介绍了归一化卷积神经网络 (NC 神经网络)，一种具有更大通用性的新方法，适用于各种卷积神经网络 (CNN) 任务。类似 BN，NC 通过标准化响应于卷积滤波器的切片输入来平滑损失景观。然而，与传统的在卷积前后应用的规范化方法不同，我们的方法直接在卷积过程中对 `im2col` 矩阵的行进行归一化。这一根本转变意味着 NC 本身就在卷积过程内运作，使其区别于现有的归一化技术。因此，NC 具有通用属性，可以应用于各种基于 CNN 的任务中，包括目标检测、实例分割等。为了展示 NC 的有效性，我们提供了理论和实证分析。我们的主要贡献是：

- **理论洞察。**理论洞察：我们证明 NC 减少了损失函数和梯度函数的利普希茨常数，导致深层卷积网络中的损失景观更加平滑。
- **经验验证。**实验结果表明，在图像分类、目标检测和分割任务中，使用 NC 的网络性能优于使用组归一化 (GN) [19] 的网络。
- **生成模型增强。**我们证明了 NC 相比位置归一化在生成模型中提供了显著的改进。为了进一步确立 NC 在视觉任务中的通用性，我们进行了广泛的实验，包括：

图像分类在多个数据集上，对象检测和实例分割在 COCO 数据集 [14] 上，语义分割在 PASCAL VOC[2] 上，以及图像生成在各种基准数据集 [21] 上。

2 相关工作

规范化方法自深度学习早期就被广泛使用。例如，局部响应规范化(LRN)被用于 AlexNet[11]。后来引入了批量规范化 (BN)，以缓解内部协变量偏移问题，从而提高了训练稳定性和收敛速度。BN 对于大批次尺寸特别有效，但当批次大小减小时，其性能往往会下降。

为了解决这一限制，提出了各种不依赖批次维度的标准化技术。层归一化 (LN) 沿通道维度对特征进行归一化，而实例归一化 (IN) 则对每个样本独立应用归一化。在这些方法中，组归一化 (GN) [19] 对于微批处理尺寸表现出高度有效性。GN 通过将通道划分为 G 个组，并在每个组内应用层归一化来泛化标准化。位置归一化 (PONO) [13] 独立地在各通道上对每个空间位置的特征进行归一化，与其它归一化技术相比，捕获输入图像更粗略的结构信息。除了这些方法外，还有许多其他的归一化方法继续被开发出来。

大多数现有的归一化技术可以如下公式表示。给定一个输入张量 $X \in \mathbb{R}^{B \times C \times H \times W}$ ，其中 B 表示批次大小， C 表示通道数量， $H \times W$ 表示空间维度，归一化沿着特定的轴进行。一般的公式是：

$$X'_K = \beta \frac{X_K - \mu_{X_K}}{\sigma_{X_K}} + \gamma,$$

其中 K 表示执行归一化的轴。可学习的参数 β 和 γ 应用于剩余的轴。例如，BN 跨 $K = B, H, W$ 进行标准化，并在 $\mathbb{R}^C$ 中学习 β 和 γ 。基于这一公式，出现了多种利用不同策略来处理 β 和 γ 的技术。值得注意的是，一些利用风格信息的方法，如条件实例归一化 (CIN) 和自适应实例归一化 (AdaIN) [6, 1] 已被研究。PONO 进一步采用了一种称为矩量快捷的技术，有效地捕捉结构信息 [13]。

规范化技术已知可以平滑非凸损失景观，从而增强训练稳定性和收敛性。我们强调 NC 表现出类似的效果，通过标准化对应于卷积核的区域来减少非凸性同时保留结构信息。

3 标准化卷积网络

批标准化 (BN) 通过显著平滑损失景观 [18]，从根本上影响网络训练，导致更稳定和更快的训练过程。这种影响扩展到最近引入的规范化方法，权重标准化 (WS) [15]。虽然 BN 主要考虑激活函数的 Lipschitz 常数，但 WS 侧重于权重。由于与这些权重相互作用的输入也决定了权重的梯度，因此 WS 同样有助于平滑损失景观。

基于这个想法，我们提出每个对应于卷积核的切片输入的标准化可以同样平滑损失景观。因此，我们认为 NC 可以被视为 WS 的对偶。

3.1 归一化卷积

考虑一个将偏置项设置为零的标准卷积层：

$$y = W * x$$

其中 $W \in \mathbb{R}^{O \times I}$ 表示权重， $x \in \mathbb{R}^{I \times HW}$ 表示层中的输入。符号 $*$ 表示卷积操作，在实践中对应矩阵乘法。对于 $W \in \mathbb{R}^{O \times I}$ ， O 表示输出通道的数量，而 I 表示每个输出通道内核区域

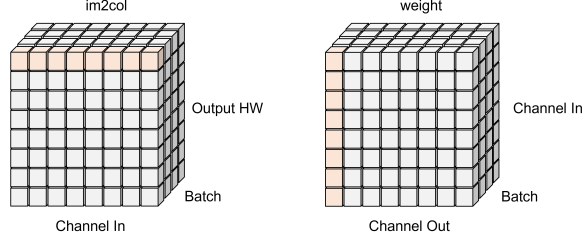


Figure 1: 标准化卷积层图。粉红色区域中的 im2col 矩阵的行被标准化，相应的权重区域也可以被标准化。

内的输入通道数量。类似地，对于 $x \in R^{I \times HW}$ ，术语 HW 表示输出宽度和输出高度的乘积。NC 按照以下方式对 im2col 矩阵 x 进行标准化：

$$\hat{x}_{i,k} = \frac{x_{i,k} - \mu_k}{\sigma_k + \epsilon},$$

其中

$$\mu_k = \frac{1}{I} \sum_{i=1}^I x_{i,k}$$

$$\sigma_k = \sqrt{\frac{1}{I} \sum_{i=1}^I (x_{i,k} - \mu_k)^2}.$$

在对 im2col 矩阵进行归一化后，可以通过矩阵乘法计算输出。值得注意的是，仿射变换可以在卷积操作之前或之后应用。在我们的实验中，我们在卷积之后应用仿射变换。

3.2 标准化卷积的梯度

我们将演示 NC 有助于平滑损失景观。涉及的计算与 [15] 中的紧密相关，并在该论文中进行了更详细的解释。具体来说，我们展示了相对于 x 优化 L 比在 $\hat{x}$ 上优化 L 具有更小的 Lipschitz 常数，在损失函数和梯度方面都是如此。函数 f 的 Lipschitz 常数是任何满足以下条件的值 A ：

$$|f(x_1) - f(x_2)| \leq A \|x_1 - x_2\|$$

对所有 x_1, x_2 成立。

对于神经网络， f 对应于损失函数 L 或其梯度 $\nabla_x L$ 。损失及其梯度的较小 Lipschitz 常数意味着在训练过程中损失和梯度的变化更加可控，允许优化器采取更大的步长，从而加速学习。

规范化过程包括两个步骤。第一步通过减去每行的平均值来中心化 im2col 矩阵的每一行，第二步通过除以标准差来进行标准化。我们将此过程重写为向量形式如下：

$$x_{:,k}^{\cdot} = x_{:,k} - \frac{1}{I} \mathbf{1} < \mathbf{1}, x_{:,k} >$$

$$\hat{x}_{:,k} = x_{:,k}^{\cdot} / \left(\sqrt{\frac{1}{I} < \mathbf{1}, x_{:,k}^{\cdot \circ^2} >} \right)$$

其中 $<, >$ 表示点积， $\circ^2$ 表示哈达玛积。

相应的梯度向量为：

$$\nabla_{x_{:,k}} L = \frac{1}{\sigma_{x_k}} (\nabla_{\hat{x}_{:,k}} L - \frac{1}{I} \langle \hat{x}_{:,k}, \nabla_{\hat{x}_{:,k}} L \rangle \hat{x}_{:,k})$$

$$\nabla_{x_{:,k}} L = \nabla_{\hat{x}_{:,k}} L - \frac{1}{I} \mathbf{1} \langle \mathbf{1}, \nabla_{\hat{x}_{:,k}} L \rangle$$

为了分析损失的 Lipschitz 常数，直接计算得出：

$$\| \nabla_{x_{:,k}} L \|^2 = \frac{1}{\sigma_k} \left(\| \nabla_{\hat{x}_{:,k}} L \|^2 + \frac{1}{I^2} \langle \hat{x}_{:,k}, \nabla_{\hat{x}_{:,k}} L \rangle^2 (\langle \hat{x}_{:,k}, \hat{x}_{:,k} \rangle - 2I) \right) \quad (1)$$

。由于标准化过程，我们有 $\| \hat{x}_{:,k} \|^2 = I$ 。因为 NC 是一个卷积过程，随后对对应于 k 的 `im2col` 矩阵的每一行进行归一化，所以 $\frac{1}{\sigma_{x_k}}$ 的影响被中和了。因此，梯度范数的主要影响是减少项： $\frac{1}{I} \langle \hat{x}_{:,k}, \nabla_{\hat{x}_{:,k}} L \rangle^2$ 。

接下来，类似的计算显示，

$$\| \nabla_{x_{:,k}} L \|^2 = \| \nabla_{\hat{x}_{:,k}} L \|^2 - \frac{1}{I} \langle \mathbf{1}, \nabla_{\hat{x}_{:,k}} L \rangle^2 \quad (2)$$

因此，(2) 对利普希茨常数的影响也是减少。

需要注意的是，第 k 层的损失梯度是通过损失相对于输入的梯度和第 k 层激活函数对其输入变量的梯度进行矩阵乘法获得的。因此，理解损失梯度如何随输入变量变化至关重要。总结 (1) 和 (2)，可以说 NC 使损失景观更加平滑。为了验证这一效果在现实世界中的表现，我们在 ImageNet[17] 上训练的 ResNet-50[4] 和 CIFAR-10/100 上的 ResNet-18 进行了案例研究。之后，还在多个著名数据集上对目标检测、语义分割和图像生成任务进行了实验。

总结 (1) 和 (2)，我们得出结论 NC 有助于平滑损失景观。为了验证这一效果在实际环境中的表现，我们在 ImageNet[17] 上训练的 ResNet-50[4] 和在 CIFAR-10/100 上训练的 ResNet-18 进行了案例研究。另外，在几个知名数据集上的对象检测、语义分割和图像生成任务中也进行了额外的实验。

3.3 CUDA 内核实现

使用 PyTorch 的伪代码如算法 1 所示。运行实验时，使用现有的 PyTorch 操作（例如通过 `unfold` 手动构造 `im2col` 矩阵）会导致显著的减速。为了解决这个问题，我开发了一个自定义的 CUDA 内核。实现可以在 GitHub 上找到：¹

4 实验

4.1 图像分类在 ImageNet、CIFAR-10 和 CIFAR-100 上

ImageNet 数据集是一个大型图像数据集，包含大约 128 万张训练样本和 5 万张验证图像，涵盖 1000 个类别。我们在微批次设置下进行了一次简单的比较，对比了使用组归一化 (GN) 的 ResNet-50 与使用 NC 的 ResNet-50，在 ImageNet 的标准训练设置下。由于 NC 采用的是不依赖于批次的归一化过程，因此我们特别将其与已证明优于其他不依赖于批次的归一化方法如层归一化和实例归一化的组归一化进行了比较。对于所有模型，GN 中的组数均设定为 32。

¹<https://github.com/kimdongsuk1/NormalizedCNN>.

Algorithm 1 规范化卷积层伪代码，类似 PyTorch

```
def NCon2d(inputs, in_channel, out_channel, kernel_size):  
  
    h,w = output_height,output_width  
    flatten_weights = flatten(weight,out_channel,kernel_size)  
  
    im2col_inputs = unfold(x,kernel_size,channel)  
    mean = im2col_inputs.mean(axis=-1)  
    std = (im2col_inputs.var(axis=-1)+epsilon).sqrt()  
    im2col_inputs = (im2col_inputs - mean) / std  
  
    output = im2col_inputs @ flatten_weights.  
  
    return output.reshape(-1,h,w,out_channel)
```

Table 1: 图像分类任务的实验结果

Dataset	Model	Method	Top-1
ImageNet	ResNet-50	GN	24.95
		NC	24.24
		GN+NC	23.81
CIFAR-10	ResNet-18	GN	10.14
		NC	8.52
		GN+NC	6.98
CIFAR-100	ResNet-18	GN	34.65
		NC	27.20
		GN+NC	26.15

所有实验均使用 PyTorch 实现，其中 NC 作为自定义层在 PyTorch 中实现。所有模型的训练批处理大小设置为 2。

尽管 CIFAR-10 数据集比 ImageNet 小，但它是在机器学习研究中最广泛使用的数据集之一。CIFAR-10/100 数据集包含分别跨越 10 类和 100 类的 60,000 张 32×32 彩色图像。我们在 CIFAR-10/100 上使用 ResNet-18 进行相同的实验。在这种情况下，我们使用没有动量和权重衰减的标准 SGD。训练批处理大小再次设置为 2，所有模型都使用 ReLU 激活函数。我们应用基本的数据增强技术，包括水平翻转和最多将图像尺寸随机平移 10% 的操作。所有模型训练 75 个周期。

此外，我们进行了实验以分析将 NC 与现有归一化方法结合的效果。具体来说，我们将带 BN/GN 的 ResNet-18 与带 BN/GN 和 NC 的 ResNet-18 进行比较。虽然在批量归一化的案例中最终准确率和验证损失值没有显著差异，但我们观察到初始训练阶段的学习速度更快，正如预期。然而，在微批次设置下，我们观察到了更明显的准确性改进。详情见图 2 和表 1。

在所有实验中，NC 一致优于 GN。由于 NC 实质上是一个卷积层，我们也评估了其与其他规范化技术结合的性能。结果显示，当与两种代表性的规范化方法 BN 和 GN 一起使用时，NC 进一步提高了性能。

4.2 目标检测和语义分割

微批量训练在目标检测和分割任务中常用。我们使用预训练的 ResNet-50 模型与 GN 和 NC 在 COCO-2017 上进行对象检测实验，并在 PASCAL-VOC 2012 上进行语义分割实验。在所有实验中，批处理大小设置为每 GPU 2 张图像，并应用迁移学习。

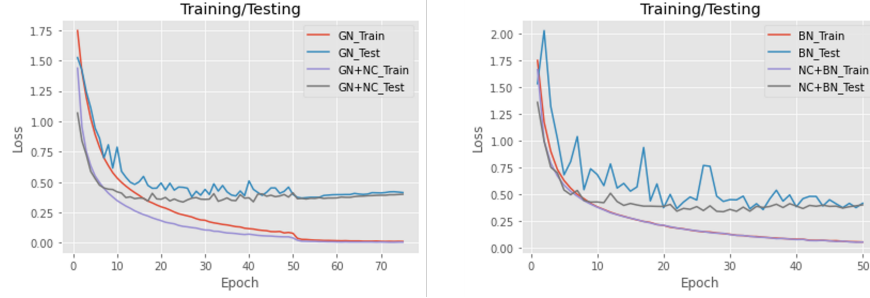


Figure 2: 图表展示了在 CIFAR-10 上的训练和验证过程中，训练和测试损失值的变化。可以观察到，训练过程更加稳定，且损失值下降速度更快。

Table 2: 对象检测和语义分割的实验结果

Tasks	DataSet	Model	AP	AP ^{0.5}	AP ^{0.75}	mIoU
Detection	COCO	GN	38.01	59.12	41.17	-
		NC	38.59	59.91	41.55	-
		GN+NC	39.01	60.34	42.07	-
Segementation	PASCAL	GN	-	-	-	74.33
		NC	-	-	-	75.90
		GN+NC	-	-	-	76.98

对于目标检测，我们采用 Faster R-CNN，遵循基于 PyTorch 的标准 Mask R-CNN 框架²。模型配备了一个特征金字塔网络 (FPN)、一个 4 卷积-1 全连接的边界框头，以及一个 1E 的学习率计划。我们使用平均精度 (AP)、AP^{0.50} 和 AP^{0.75} 来评估边界框检测的性能。

对于语义分割，我们使用具有相同预训练 ResNet-50 骨干网络的 DeepLabV3。评估在 21 个不同的类别上进行，遵循³中建立良好的实验设置。批量大小设置为 2，并使用平均交并比 (mIoU) 指标来评估模型性能。

如表 3 所示，两项任务的结果均表现出与图像分类中观察到的趋势相似的现象，进一步证实了 NC 与现有归一化方法结合的好处。这些发现表明，NC 增强了基于卷积的知名任务的性能。未来的工作将集中在验证这些效果在各种骨干架构上的表现，并将分析扩展到实例分割。

4.3 比较结构效应对生成模型的影响

与位置归一化的关联 NC 在对应的核大小为 1E1 的二维卷积过程中等同于位置归一化 (PN)。因此，将 NC 与 PN 进行比较是很自然的。值得注意的是，NC 被设计成更好地保留结构信息，因为它对相应的权重通道进行归一化，确保关键的空间特征保持完整。

深度生成模型 大多数深度生成模型采用编码器-解码器架构。自从生成对抗网络 (GANs) [3] 引入以来，已经开发出众多变体，取得了显著成果。例如，Zhu 等人 [21] 提出了输入图像和输出图像之间的图像到图像翻译，而 Huang 等人 [8] 实现了从输入图像 A 到输出图像 B 的风格转换。U-Net[16]，一个著名的编码器-解码器模型，在分割任务中表现出色。U-Net 的核心原则也被有效地应用于各种生成模型。

²<https://github.com/facebookresearch/maskrcnn-benchmark>

³<https://github.com/VainF/DeepLabV3Plus-Pytorch>

Table 3: 使用 CycleGAN 和 Pix2pix 进行双向图像生成的实验结果

Dataset	Baseline	Method	FID (A $\rightarrow$ B)	FID (B $\rightarrow$ A)	LPIPS (A $\rightarrow$ B)	LPIPS (B $\rightarrow$ A)
Maps	CycleGAN	PONO+MS	52.45	64.13	0.3317	0.175
		NC+MS	49.57	59.78	0.327	0.177
Horse2Zebra	CycleGAN	PONO+MS	71.70	136.61	0.757	0.730
		NC+MS	67.54	133.81	0.731	0.763
Maps	Pix2pix	PONO+MS	57.39	62.39	0.334	0.171
		NC+MS	52.85	60.01	0.329	0.171
Day2Night	Pix2pix	PONO+MS	200.69	188.49	0.6198	0.5781
		NC+MS	195.01	179.97	0.624	0.5356



Figure 3: 一组测试图像被展示出来。前两张图像是原始图像，第三张图像是使用 PONO 生成的，最后一张图像是使用 NC 产生的。

时刻捷径 位置归一化 (PONO) 在不同层捕获结构签名信息。利用这一想法，提出了矩简捷 (MS) 技术。MS 将从编码器层提取的位置矩信息 μ 和 σ 传递到反卷积解码器层。该技术可以独立于归一化应用。

为了比较结构特征提取能力，我们使用两个基线模型评估 NC (+MS) 和 PONO (+MS): CycleGAN 和 Pix2Pix。唯一的区别在于是否使用了归一化或 NC。对于所有模型，我们都遵循官方实现⁴中提供的相同实验设置。此外，我们采用了连接的跳层连接，其中编码器激活直接与解码器激活连接，类似于 U-Net。有关更多架构细节，请参阅 [13]。

评估指标 我们使用两个广泛采用的评估指标：

- **Fréchet 初始距离 (FID) [5]** 此指标计算目标域中生成图像与真实测试图像分布之间的距离。它基于从在 ImageNet 上预训练的 Inception-V3 模型中提取特征的高斯分布。FID 提供了生成图像与真实数据分布匹配程度的估计。
- **学习感知图像块相似性 (LPIPS) [20]** 该度量衡量了生成图像与目标图像之间的感知相似性。LPIPS 使用从 AlexNet[11] 提取的特征进行计算。已经显示 [20] 它与人类对图像相似性的感知判断有很强的相关性。

数据集 我们在三个数据集上进行了实验：

- **马 2 斑马** 该数据集包含 1,067 张马的图像和 1,334 张斑马的图像用于训练，以及 120 张马的图像和 140 张斑马的图像用于测试。数据集是未配对的，并源自 ImageNet。
- **映射 [21]** 该数据集包括每个领域 1,069 张训练图像和 1,098 张测试图像。
- **日转夜 [12]** 该数据集包含 17,823 张自然场景图像用于训练和 2,287 张图像用于测试。

遵循官方的 CycleGAN 和 Pix2Pix 实现，我们将 PN 集成并替换为 NC。前两个数据集用于 CycleGAN，而后两个数据集用于 Pix2Pix。

⁴<https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>



Figure 4: 图像通过 CycleGAN 获得。第一幅图代表原始图像，第二幅图使用 PONO 生成，第三幅图使用 NC 产生。值得注意的是，从 NC 获得的图像包含更多详细信息，这一点可以直观观察到。

所有模型都训练了 200 个周期，FID 和 LPIPS 是在最终结果上计算的。在评估 NC 时，我们仅将标准卷积层替换为我们的 NC 层，同时保持与 PONO 相同的层位置。在所有实验中，NC 相较于现有方法均表现出性能提升。表 3 所示的结果证实了 NC 增强了生成模型的表现。此外，在图 3 和图 4 中可以观察到定性上的改进。

5 结论与未来工作

在本文中，我们介绍了归一化卷积 (NC)，这是一种受权重标准化 (WS) [15] 在训练过程中平滑效果启发的新颖卷积范式。我们的方法可以被视为 WS 的对偶版本，符合微分卷积算子的基本属性。具体来说，在卷积操作中，如果一个函数是平滑的，则整体卷积倾向于继承该函数的平滑性特性，即使另一个函数本身不是平滑的也是如此。通过理论分析，并遵循 [15] 中的计算，我们证明 NC 也减少了 Lipschitz 常数，进一步促进了训练稳定性。

实证结果表明，NC 的表现优于组归一化 (GN)，使其成为微批次图像分类及相关任务的有力候选方法。基于 [15] 中的研究发现，我们假设 WS+NC 可以作为此设置下的最先进的方法。然而，我们的实验在将 WS 与 NC 结合时并未带来显著改进，这表明需要进一步调查。

在批次独立归一化技术中，位置归一化 (PN) [13] 与 NC 有相似之处，因为两者在使用 1×1 内核时会达到一致。然而，由于内核大小并不总是 1×1 ，NC 通过直接对切片输入进行归一化提供了更加适应的方法。我们的实验表明，PN 更适合于基于 GAN 的任务，而 NC 应用于这些模型时则表现出改进。需要进一步研究以充分探索 NC 在各种任务中的潜力。

未来研究的另一个关键方向是为 NC 识别一个更合适的激活函数。虽然 SELU[10] 展现了自归一化特性，但我们的实验表明它并不是 NC 的最佳选择。通过改进初始化策略，SELU 仍可能证明是有益的，未来的工作应该探索这一可能性。

最后, 我们观察到 NC 在使用自适应梯度优化器如 Adam[9] 时表现不佳。这表明 NC 需要专门的自适应优化方法来充分发挥其优势。未来的研究应专注于开发一种针对 NC 的优化策略, 可能改善训练动态并提升整体模型性能。

总之, 我们的研究强调 NC 作为一种有前景的标准化技术, 在各个领域都有应用。然而, 为了使其与不同的架构、优化器和激活函数更好地结合, 还需要进一步的研究, 为更强大和有效的深度学习模型铺平道路。

6 致谢

此项研究获得了韩国贸易、工业和能源部 (MOTIE) 在“区域特色产业开发计划 (研发, P0002072)”下的资金支持, 该计划由韩国技术振兴院 (KIAT) 监管。

References

- [1] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *CoRR*, abs/1610.07629, 2016.
- [2] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.
- [3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [5] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017.
- [6] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *CoRR*, abs/1703.06868, 2017.
- [7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.
- [9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [10] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *Advances in neural information processing systems*, 30, 2017.
- [11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

- [12] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on graphics (TOG)*, 33(4):1–11, 2014.
- [13] Boyi Li, Felix Wu, Kilian Q Weinberger, and Serge Belongie. Positional normalization. In *Advances in Neural Information Processing Systems*, pages 1620–1632, 2019.
- [14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [15] Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. Weight standardization. *arXiv preprint arXiv:1903.10520*, 2019.
- [16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.
- [17] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.
- [18] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.
- [19] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.
- [20] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [21] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.