

# 多模态转移：应用于快速艺术风格转移的分层深度卷积网络

谭洁

21821109

tanjie951102@163.com

## 摘要 (Abstract)

将艺术风格转移应用到日常照片上是目前工业界和学术界的一件非常流行的任务。最近，离线训练已经取代在线迭代地优化，使得照片的艺术化接近实时。然而，当把高分辨率的图像直接应用到这些风格网络时，产生的艺术化图像在某些局部区域和我们想要的艺术化效果有些出入。这是由于艺术迁移的过程没有很好地把握住小而微妙的一些纹理，没有维持好艺术作品纹理的尺度。在这里，我们提出一种把图像的色彩和亮度通道表示纳入其中的多模卷积神经网络，该神经网络通过将对不同尺寸图像的多种loss结合起来，进行层级化的艺术化。和现在最好的网络相比，我们的网络通过进行更加复杂的离线训练可以达到实时的艺术化。通过从多个尺寸进行风格和纹理线索的提取，我们不仅能转移大尺度、明显的风格线索，也能提取微妙、精细的风格线索。也就是说，我们的方法能产生在视觉上让人更满意的结果，也更接近于有多重尺度的色彩和纹理的艺术风格。

### 1. 引言

风格迁移，或者说使用另一种风格对图像进行重绘，这项工作被普遍认为是一个有挑战性而有趣的问题。最近这项工作由于Gatys等的一个影响极大地工作激起了其在工业界和学术界的热潮。在Gatys的工作中，一个预训练好的用于视觉识别的网络被用来提取一张图像的内容和风格表征，这项工作得到了让人震惊的视觉效果。但是，由于需要经过一个在线的、迭代优化过程，它的风格迁移所耗时间让人望而生畏。为了解决这个问题，可以训练一个能够进行离线训练的、带有相同loss的前馈神经网络，这样做可以产生非常接近的视觉效果。但是在应用时就只需要将图像输入前馈神经网络，进行一次推理就能够得到最终的风格迁移图。这么做的结果是一个可以快上百倍的计算算法。

尽管已有的工作通过多种不同的网络能够产生视觉效果不错的结果，其实它们有两个非常突出的缺点：1. 现有的前馈神经网络都是在固定分辨率的风格图上训练的，这样，当偏离固定尺寸的分辨率便会产生尺度上

的不匹配。例如，把一张高分辨率图像输入一个在大小为256的风格图上训练的模型时，产生的结果的纹理尺度会小于风格图的纹理尺度。2. 当前的网络很难捕捉到很多风格的高分辨率艺术作品上的微小、精细的纹理，如笔刷。现有的网络能够捕捉到抽象的、大尺度的纹理和一些容易分辨的笔触，如梵高的《星空》，尽管这些已经被证实，但是仅有这些还是不够，艺术风格所包含的内容要远超这些已经被证实的。也就是说：很多的艺术风格其特征是一些微妙、精细的笔刷和笔触。由此，我们的结论是现有的一些风格转移网络的结果对于很多类别的艺术风格而言都是不够好的。

本文中，我们提出一种全新的层级化的深度卷积网络来进行快速风格迁移。我们贡献包含4个方面：1. 我们提出了一种层级化的网络和其训练方法，通过利用多尺度的风格图，该网络既能够学习到粗略的、大尺度的纹理，也能学习到精细的笔刷等纹理；2. 我们的层级化训练方法和端到端的CNN网络架构让我们能够将多个模型融合到一个网络中，从而处理多个尺度的图像；3. 我们的模型把RGB三个颜色通道以及亮度通道利用起来进行风格迁移；4. 通过实验，我们证明我们的层级化网络能够更好的捕捉粗略的和精细的纹理模式。

我们的层级化风格迁移网络在训练时使用了多个风格loss，这些不同的loss分别对应不同的图像尺寸，这些loss通过（mixture of modalities）结合到一起。由此为了和仅仅使用一个风格loss的前馈风格迁移神经网络相区别，我们把我们的网络称之为多模迁移，而把前者称之为单模迁移。在图1中，我们把我们的多模迁移的结果和当且最好的单模迁移的结果进行了对比。图1显出了多模迁移在学习不同层级的纹理时的优越性，这些纹理包括风格、颜色、大的纹理分布和精细的笔触。须特别注意的一点是：我们的方法更加接近于艺术画的笔触。在第4节中，我们还将展示多模迁移也能用来训练一个能够使用多种风格图对一张图像进行风格化的组合模型。仅使用一个风格loss的前馈风格迁移神经网络相区别，我们把我们的网络称之为多模迁移，而把前者称之为单模迁移。在图1中，我们把我们的多模迁移的结果和当且最好的单模迁移的结果进行了对比。图1显出了多模迁移在学习不同层级的纹理时的优越性，这些纹理包括风格、颜色、大的纹理分布和精细的笔触。须特别

注意的一点是：我们的方法更加接近于艺术画的笔触。在第4节中，我们还将展示多模迁移也能用来训练一个能够使用多种风格图对一张图像进行风格化的组合模型。



图1:上排: (a) 风格是Tomas King的《At The Close of Day》，(f) 是内容图。(b) 是Gatys等人基于优化的方法的结果（由于内存限制，分辨率限制为512）。(c)，(d) 和 (e) 是由不同的前馈网络（结果分辨率均为1024）产生的结果。下排：对顶行的红色框的局部放大。可以看到，所有结果都用风格图像的颜色重新绘制。然而，更进一步看，在(c)和(d)中风格画的笔刷纹理并没有很好地捕获到。(b)中的放大区域有点模糊。与其他方法相比，我们的多模风格化(e)能够在高分辨率图像上更逼真地模拟原始艺术品的笔触。

## 2. 相关工作

### 2.1. 理解深度神经网络的表示

最近，在理解深度神经网络方面取得了开创性的工作。Zeiler和Fergus的DeconvNet方法[30]通过识别哪些图像块负责某些神经元的激活来学习如何获得某些网络输出。Yosinski等人[29]旨在通过可视化内部神经元来了解深层网络执行的计算。Mahendran和Vedaldi [19]反转了某些层的图像表示，以了解网络保留了哪些信息。后两种方法生成具有优化过程的可视化图像，其目的是用于对网络功能的感知理解。在其他情况下也采用类似的优化程序[23,20]。

基于对深度卷积网络强大表征能力的更好理解[15]，许多传统的视觉任务已得到解决，其结果也得到了更好的改善。基于优化的风格迁移就是这样一个例子。与以前的纹理合成算法不同，后者通常是非参数方法[5,28,4,11,1,16,17]，Gatys等首先提出了一种合成纹理图像的优化方法，其目标损失的计算是基于预训练的卷积神经网络的表示[6]。然后将这种纹理损失与来自Mahendran和Vedaldi [19]的内容损失相结合，以执行样式迁移任务[8]。

### 2.2. 用于图像生成的前馈网络

由于迭代优化过程，用于图像生成的基于优化的方法在计算上是昂贵的。相反，许多深度学习方法使用从神经网络计算的感知目标作为损失函数来构建前馈神经

网络以合成图像[3,9,2,22]。

快速的风格迁移取得了很好的效果，并受到了广泛关注。约翰逊等人[13]使用Gatys[8]等人定义的感知损失，提出了一种快速风格迁移和超分辨率的前馈网络。一个类似的架构---纹理网---被引入来合成纹理和风格化的图像[25]。最近，Ulyanov等人[26]证明，在快速风格化中，使用实例标准化替换前馈网络中的批标准化（BP）[12]可以显着提高生成图像的质量。在这里，我们对这种样式迁移算法做出进一步改进，以使用具有多模态的分层网络处理逐级变大的图像。此外，它允许使用多种不同的样式来重新绘制单个输入图像。

## 3. 多模态转移网络

### 3.1. 总体架构和学习方法

我们提出的网络如图2所示，由两个主要部分组成：前馈多模网络和损失网络。前馈多模网络（MT网络）是分层深度残差卷积神经网络。它由三个子网组成：样式子网，增强子网和细化子网。这些子网分别由 $\Theta_1$ ， $\Theta_2$ 和 $\Theta_3$ 参数化。在高层级，MT网络将图像 $x$ 作为输入并且被训练以生成尺寸增加的多个输出图像 $\hat{y}_k$ ,

$$\hat{y}_k = f(\bigcup_{i=1}^k \Theta_i, x)$$

然后将这些输出图像分别作为损失网络的输入，以计算每个输出图像的风格化损失。总损失是所有风格化损失的加权组合。稍后会给出损失网络和总损失的定义。

在推理时，为了在应用于较大图像时产生相同的样式效果和正确的艺术纹理比例，MT网络分层地对图像进行样式化：输入图像首先由双线性下采样层下采样到256，然后由样式子网捕获图稿整体的颜色和纹理特征。接下来，作为第一个输出 $y_1$ 的风格化结果被上采样到512并且通过增强子网得到输出 $y_2$ ，这增强了样式化强度。然后将其上采样到1024。最后，细化子网剔除局部像素化瑕疵并进一步细化结果。在这三个阶段处理之后获得高分辨率和最具视觉吸引力的结果 $y_3$ 。

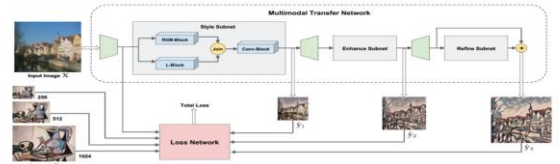


图1: 总体架构

### 3.2. 损失函数

在本节中，我们首先介绍单一的风格化损失函数，然后提出一种分层的风格化损失函数，用于训练我们的多

模态风格迁移网络。

### 3.2.1 单一风格化损失函数

类似于先前快速风格迁移工作中的损失定义[13,25]，风格化损失也来自Gatys等人[8]，即使用一个损失网络（针对对象识别优化的预训练的VGG-19网络[24]）来提取图像的表达。

定义两个感知损失以定量所生成的图像 $y_k$ 将内容图 $y_c$ 的内容与样式图 $y_s$ 的纹理和样式组合的程度（参见图3）。

**内容损失：**

内容丢失函数用于度量 $y_k$ 和 $y_c$ 之间的不相似性。设 $F_i^l(x)$ 表示应用于图像 $x$ 的损失网络的第 $l$ 层中的第 $i$ 个特征映射。内容损失是第 $l$ 层的两个特征表示之间的平方差：

$$\mathcal{L}_{content}(y_k, y_c, l) = \sum_{i=1}^{N_l} \|F_i^l(y_k) - F_i^l(y_c)\|_2^2$$

也就是说，内容丢失直接比较从相应层的特征映射，因此适合于表征空间内容相似性。

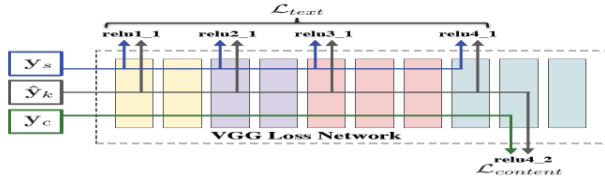


图2：损失网络

**纹理或风格损失：**

Gatys 等人提出损失网络的每一层中的特征图之间的相关性可以被视为图像的纹理表示[6,8]。这些相关性由 Gram 矩阵给出，其元素是这些特征映射之间的成对点积：

$$G_{ij}^l(x) = \langle F_i^l(x), F_j^l(x) \rangle$$

最后来自 MT 网络的每个输出  $y_k$  的风格化损失被定义为内容损失和纹理损失的加权和：

$$\mathcal{L}_S(y_k, y_c, y_s) = \alpha \mathcal{L}_{content}(y_k, y_c) + \beta \mathcal{L}_{textural}(y_k, y_s)$$

### 3.2.2 层级化的风格损失函数

多模迁移网络可以生成  $K$  个逐级增大输出结果（图2中所示的网络中  $K = 3$ ）。然后针对每个输出结果  $y_k$  计算样式化损失：

$$\mathcal{L}_S^k(y_k, y_c^k, y_s^k) = \alpha \mathcal{L}_{content}(y_k, y_c^k) + \beta \mathcal{L}_{textural}(y_k, y_s^k)$$

其中  $y_c^k$  和  $y_s^k$  分别对应内容和样式，作为子网的输入最终得到  $y_k$ 。通过训练具有不同风格尺度的子网，我们可以控制不同的子网学习不同的艺术特征类型。

由于这种风格化损失是基于整个网络的不同层的输出来计算的，因此这里不能使用总损失（例如，所有风格化损失的加权组合）来直接传播和向后更新参数。因此，采用并行方法，使得不同的风格化损失用于反向传播不同范围的层的参数。我们定义了分层风格化损失函数  $\mathcal{L}_H$ ，它是这种风格化损失的加权和：

$$\mathcal{L}_H = \sum_{k=1}^K \lambda_k \mathcal{L}_S^k(y_k, y_c^k, y_s^k)$$

其中  $\lambda_k$  是风格损失的权重。

因此，在对自然图像  $x \sim X$  的端到端学习过程

中，用参数  $\Theta_k$  表示的每个子网都被训练以最大限度地减少从后面的输出  $y_i$  ( $i \geq k$ ) 计算的并行加权风格化损失（后者意味着它在前馈方向上位置更靠后），如下式：

$$\Theta_k = \arg \min_{\Theta_k} E_{x \sim X} \left[ \sum_{i=k}^K \lambda_i \mathcal{L}_S^i(f(\bigcup_{j=1}^k \Theta_j, x), y_c^i, y_s^i) \right]$$

从上式我们可以看到，即使所有这些子网都是为不同的目的而设计的，但它们并不完全独立。前子网也有助于减少后面子网的损失函数。因此，可以为后面的子网寻找较浅的 CNN 结构，这节省了计算存储和运行时间。

### 3.3. 网络架构

单模迁移网络的一个主要缺点（例如[13,25]）是训练网络时的单一模限制了捕获的样式细节的范围。由于它是使用风格图像的一个特定比例训练的，因此在训练期间我们需要选择是否学习粗糙纹理或精细笔触。也就是说，它以牺牲另一个为代价来学习一个。

为了解决这个问题，我们设计了层次结构，用不同尺度的样式图像训练不同的子网，以学习不同层次的纹理线索。这种设计使得图像能够以逐级增加的分辨率使用不同级别的样式进行迁移。此外，由于所有这些子网都组合成一个网络并经过层次训练，后面的子网也能够增强和完善前面的子网，使我们的网络成为提高效率和鲁棒性的协作方案。

我们已经尝试了几种具有不同层次结构和不同内部结构的架构。这里我们介绍图2所示的网络的一般架构，从我们的实验所得它具有最好的风格化效果。

如前所述，多模式传输网络由三个可学习的子网，样式子网，增强子网和细化子网组成，每个子网遵循固定的双线性上采样/下采样层。请注意，增强子网和精简子网之间的上采样层仅在测试时插入，因此在训练期间，精炼子网的输入仍然是 512 的大小，这极大地减少



了所需的内存并加快了训练过程。这些网络的显著特征解释如下。

### 3.3.1 样式子网

**亮度-色彩联合学习：**为了更好地解决保留小精细纹理的问题，我们的网络利用了颜色和亮度通道的表示，因为视觉感知对亮度的变化比对颜色更敏感[27, 11, 7]。我们将亮度通道与RGB彩色图像分开，并使用两个独立的分支（RGB-Block和L-Block）来区别地学习它们的表示。然后，从两个分支计算的特征图拼合在一起，并由随后的Conv-Block进一步处理。

RGB-Block包括三个跨步卷积层（分别为 $9 \times 9$ ,  $3 \times 3$ ,  $3 \times 3$ ，后两个用于下采样）和三个残差块[10]，而L-Block具有类似的结构，除了卷积的深度是不同的。Conv-Block由三个残差块组成，两个用于上采样的resize-convolution层和最后一个 $3 \times 3$ 卷积层以获得输出RGB图像 $y$ 。所有非残差卷积层之后都是实例归一化[26]和ReLU非线性。我们风格子网的一部分是基于工作[13, 22]设计的。

这里使用最近邻插值上采样层和称为大小调整卷积层（resize-convolution）的卷积层而不是反卷积来避免生成图像的棋盘伪像[21]。

### 3.3.2 增强子网和细化子网

虽然样式子网旨在使具有较大纹理失真的输入图像样式化以匹配样式图的样式，但我们发现，对很多类型的风格而言，想要在保留内容图的整理性的情况下，通过调整纹理和内容权重以实现样式转移是比较难的。因此，我们允许样式子网执行纹理映射，着眼于保留内容，并训练单独的增强子网，并赋予纹理较大的权重以进一步增强样式化。图4说明了每个子网的具体作用。显然，样式子网会严重改变颜色和纹理，但增强子网也会在添加更多细节时对纹理映射做出很大贡献。细化网进一步细化并在最终结果中添加更多细节。

## 4. 实验结果与分析

### 4.1. 训练细节

多模态风格迁移网络是在微软的COCO[18]数据集上的子数据集上训练的，它包含32059张图片（宽和高都 $\geq 480$ ），我们将其裁剪调整大小为 $512 \times 512$ 。使用Adam优化器[14]来训练模型，每个批量为1训练10000次，学习率初始化为 $1e-3$ ，每2000次迭代减少0.8倍。内容losses使用VGG-19的relu4\_2层计算，纹理losses由relu1\_1, relu2\_1, relu3\_1和relu4\_1。内容权重重置为1，而纹理权重取决于不同的风格。因为一个通用的纹理和内容的比例不适用于所有的艺术风格。对于风格losses的权重，我们设置为 $\lambda_1: \lambda_2: \lambda$

$3=1:0.5:0.25$ 。原因是在训练过程中，先前的子网络参数被更新来和现在的和后续的风格losses合作。后续的损失需要更小的权重为了不支配先前网络的优化过程。实验表明，结果随着 $\lambda$ 的调整而变得鲁棒。需要在NVIDIA GTX1080 花费大约1小时来训练一个分层网络。模型大小接近35MB。



图3: (a)是内容图，(b)是风格图，(c) (d) (e)是三个子网的输出 $y_1, y_2, y_3$ ，分辨率分别为256, 512, 1024。最后一行是相互直接的绝对差值图：(f)是内容图和 $y_1$ ，(g)是 $y_1$ 和 $y_2$ ，(h)是 $y_2$ 和 $y_3$ 。

### 4.2. 和先前的单一转移网络比较

前面提到，单一的风格转移是一个前向的风格转移网络使用单一的风格loss，而多模态的转移网络是一个分层的网络，使用多个风格losses和混合多模态（颜色和光度）。这里我们分离我们的多模态迁移网络的子网络作为一个单模态网络来和先进的网络[13][26]比较。所有的这三个网络在大小为256的图像上训练。我们的模型的子网络在获取纹理细节上比其他的网络要更好，如图5所示。第一个例子，Ulyanov等人的结果在颜色上比风格图像要暗，而纹理规模上Johnson等人的结果没有很好地匹配纹理图像。我们的自风格网络好像在两个方便都做得更好。在第二个例子中，我们的风格子网络在处理小和细致的纹理上要更好些。



图5: 我们的子网络(b)和其他单一迁移网络(c)和(d)的比较。

### 4.3. 单模态转移和多模态转移在高分辨率图像的区别

我们在不同的艺术风格上来测试我们的方法。在图6中，我们比较了在高分辨率图像（ $1024 \times 1024$ ）在多模态迁移和单模态转移的不同。实验结果如图6表明，相比于单模态，多模态迁移的结果和原始的风格图片在粗的纹理和细致的粉刷上更相似。而单模态迁移在风格图像大小为256图像上会出现错误匹配的结果。另外，

单模态的风格迁移在分割大小为 1024 很难学到模糊和细致的粉刷。下图第一列是我们的风格图像，第二三四列分别是我们训练的不同风格模型生成的结果，从低分辨率 256 到高分辨率 1024 再到多层分辨率结果。

#### 4.4. 多模态的风格迁移使用更多的风格

我们的多模态风格迁移允许之前一个不可能实现但有趣的应用：他可以在多个风格图片中训练，最终的风格图片融合测试集的内容图片和一个粗的纹理图片和一个细致的粉刷图片风格。如图 7 所示。

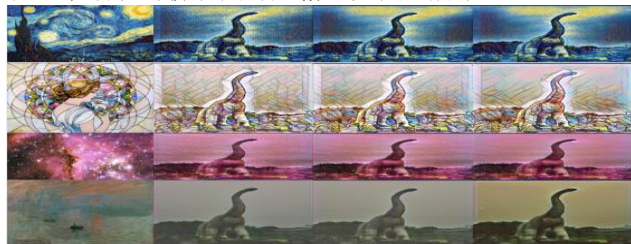


图 6：不同风格模型的生成结果

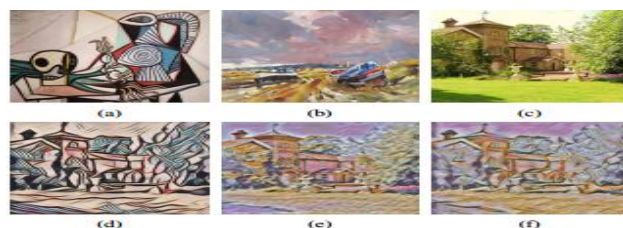


图 7：使用两种风格进行多模态转移模型在风格a和b训练。c是测试的图像，f是最终的风格化的结果，有着a的纹理和从b学到的小和细致的粉刷。通过对比，我们给出了模型从单一风格迁移后的结果d和e。

#### 4.5. 处理速度和使用内存比较

我们定量的比较我们的多模态迁移网络（MT Net）和其他的单模态的迁移网络（如 Johnson Net）。我们还重现一个深度单模态网络（称作 DS Net）作为比较，它和 MT Net 具有相同的结果。我们使用 1000 张生成结果来比价测试速度。如表 1 所示：结果表明，虽然 MT Net 比 Johnson Net 超过两倍深，但在生成高分辨率图像时运行速度和内存使用非常接近 Johnson Net。而在低分辨率下却能从分层迁移过程中获益更多。因此，多模态风格迁移很适合在现实场景的应用，如要求高分辨率的图像，因为他可以生成接近目标艺术风格的图像却消耗更少。

表 1：在 1024\*1024 图像上速度和内存使用比较

Network	Test Time	Memory Usage
MT Net	0.54s	3100 MB
Johnson Net	0.42s	2400 MB
DS Net	0.63s	6700 MB

#### 5. 结论

在这里，我们提出了一种分层训练方案（多模式转换），用于快速风格转换，以多种尺度学习艺术风格线索，包括颜色，粗糙的纹理结构和精致，精致的笔触。该方案解决了纹理尺度不匹配问题，并在高分辨率图像上生成更具视觉吸引力的风格化结果。

在未来，我们计划调查其他损失，以更好地捕捉不同规模的艺术风格。我们还想探索耗费更少内存的备用损耗网络，以将我们的方案扩展到更大的图像上。

#### References

- [1] N. Ashikhmin. Fast texture transfer. IEEE Computer Graphics and Applications, 23(4):38–43, 2003.
- [2] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In Advances in neural information processing systems, pages 1486–1494, 2015.
- [3] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1538–1546, 2015
- [4] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pages 341–346. ACM, 2001
- [5] A. A. Efros and T. K. Leung. Texture synthesis by nonparametric sampling. In Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, volume 2, pages 1033–1038. IEEE, 1999
- [6] L. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In Advances in Neural Information Processing Systems, pages 262–270, 2015.
- [7] L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman. Preserving color in neural artistic style transfer. arXiv preprint arXiv:1606.05897, 2016
- [8] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2414–2423, 2016
- [9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems, pages 2672–2680, 2014
- [10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.

- [11] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340. ACM, 2001.
- [12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [13] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.
- [14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [16] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003.
- [17] H. Lee, S. Seo, S. Ryoo, and K. Yoon. Directional texture transfer. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, pages 43–48. ACM, 2010.
- [18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.
- [19] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 5188–5196. IEEE, 2015.
- [20] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436. IEEE, 2015.
- [21] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. <http://distill.pub/2016/deconvcheckerboard/>, 2016.
- [22] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [23] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
- [24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [25] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Int. Conf. on Machine Learning (ICML)*, 2016.
- [26] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.
- [27] B. A. Wandell. *Foundations of vision*. Sinauer Associates, 1995.
- [28] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488. ACM Press/Addison-Wesley Publishing Co., 2000.
- [29] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.
- [30] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.