天津大学

《计算机视觉技术实验报告》

题目：Remastering (SIGGRAPH Asia 2019)

学　　院　智能与计算学部

专　　业　计算机科学与技术

年　　级　2019

组长姓名　张明君（留学生）

组长学号　6319000359

指导教师　林迪

2022年 4 月 24 日

摘　要

自 19 世纪后期电影发明以来，令人难以置信的小时数的电影已被记录和发行。但是，除了视觉伪影和低质量的当时的电影技术，许多早期的重要作品历史价值已遭受退化或丧失。鉴于此类重要电影的相似性，它们的修复工作很复杂，最初的努力就是恢复在物理层面上拍摄。之后，将内容传输到数字媒体，通过消除噪音和除了为胶片镜框增添色彩外，还有一些人工制品。然而，这种重新制作过程需要大量的时间和金钱，目前由专家手动完成，单部电影的成本在数十万至数百万之间美元。在这种情况下，拥有大量已存档的退化的旧视频的庞大行业，例如出版商，电视和印刷行业，都显示出对高效修复技术的巨大需求。在这项工作中，我们提出了一种半自动方法，用于修复已转换为数字数据的旧黑白胶片。

修复旧电影并不像使用降噪一样简单算法，然后采用流水线方式的着色方法：噪声和着色过程相互交织，并相互影响其他。此外，大多数老电影都存在模糊和低分辨率的问题，为此，提高清晰度也变得很重要。我们提出了一个完整的流水线来重新制作黑白电影，该流水线由几个可训练的组件组成，这些组件在单个端到端框架中进行训练。通过精心的数据创建和扩充方案，我们能够训练模型以进行重新制作不仅可以消除噪音和增加色彩，还可以提高分辨率和清晰度，并改善对比度具有时间一致性。

总结一下，我们的贡献如下：（1）第一个单曲重新制作老式电影的框架，（2）参考源注意（source-reference attention）可以处理任意数量的参考图像，（3）基于实例的薄膜降解模拟方法胶片修复的培训数据，以及（4）进行深入评估与现有方法相比，结果令人满意，基线。

目 录

1. 实验内容与目的

图 1.1 复古电影的结果 “A-Bomb Blast Effects” (1952)

我们的方法能够在一次处理中仅使用 6 张参考彩色图像来重新制作 700 帧视频步。图 1.1 展示：第一行显示输入视频的各种帧，第二行显示恢复的黑白帧，第三行显示最终的彩色输出，和最右边我们显示参考彩色图像。

使用 source-reference attention，我们的模型会自动将相似区域匹配到参考彩色图像，并使用 self-attention with temporal convolutions，能够强制时间一致性。 我们的方法能够恢复嘈杂和模糊的输入，然后，使用少量手动着色的参考图像，我们可以获得时间上一致的自然彩色视频。

老式胶片的重新制作包括多种子任务，包括超分辨率，噪声消除和对比度增强，旨在将退化的胶片介质恢复到其原始状态。另外，由于时间的技术限制，大多数老式胶片要么以黑白记录，要么具有低质量的颜色，因此必须进行着色。在这项工作中，我们提出了一个单一的框架来半交互式地解决整个重新制作任务。我们的工作基于时空卷积神经网络（temporal convolutional neural networks），其注意力机制在具有数据驱动的劣化模拟的视频上进行训练。我们提出的源参考关注度（source-reference attention）允许该模型处理任意数量的参考彩色图像，以使长视频着色，而无需在保持时间一致性的情况下进行分段。定量分析表明，我们的框架优于现有方法，并且与现有方法相比，随着更长的视频和更多的参考彩色图像，我们框架的性能得以提高。

2. 实验相关知识

2.1. Denoising and Restoration

降噪和复原的经典方法之一是块匹配和 3D 过滤（BM3D）算法家族，它们基于转换域中的协作过滤。尽管可以消除的噪声模式类型非常有限，但是这些方法对图像和视频均具有广泛的适用性。除了消除噪声外，还使用 BM3D 算法探索了其他与恢复相关的应用，例如图像超分辨率和去模糊。

最近 ， Convolutional Neural Networks 适用于降噪型应用，尤其是单张图像。但是，这些通常假设简单的加性高斯噪声，模糊或 JPEG 解块 (JPEGdeblocking)，或应用于特殊任务例如容易创建的 Monte Carlo 渲染去噪监督训练数据。基于光流的视频扩展还提出了变压器和变压器网络。但是，恢复旧胶卷不仅需要消除高斯噪声或模糊，还需要消除胶卷。可能都是局部的伪影，会影响图像的一小部分区域，或整体，影响整个画面的对比度和亮度，如图 2.1 所示。为此，必须创建更高的质量 和我们在方法中建议的逼真的胶片噪音。

(a) Synthetic noisy images. (b) Vintage black and white movies from the early 20th century.

图 2.1 Comparison between denoising and restoration tasks

（a）为去噪任务生成的合成图像的示例。 的上排显示原始图像，下排显示添加了高斯噪声的图像。

（b）需要修复的老式胶卷示例。 这些老电影遭受许多劣化问题的困扰，例如胶片颗粒噪声，划痕，潮湿，发粘和对比度流血，这使它们很难恢复其原始质量。

2.2. Colorization

黑白图像的彩色化是一个不适定的问题没有单一的解决方案。大多数方法都依赖根据用户输入的形式，可以是涂鸦形式，类似于要着色的图像的参考图像或互联网查询。虽然大多数传统方法都专注于使用输入灰度图像和用户提供的提示或参考图像来解决优化问题，但最近的方法选择利用大型数据集并采用基于学习的模型（例如卷积神经网络（CNN））进行着色自动图像。与基于优化的方法类似，基于 CNN 的方法已扩展为既可以将用户输入同时显示为涂鸦，又可以处理单个参考图像。我们的方法与现有的基于 CNN 的方法有关，将彩色扩展到视频和任意数量的参考图像，以及执行视频还原操作。

与当前工作相关的是递归神经网络(Recursive Neural Network)（RNN）为视频着色的方法。他们通过将颜色从初始的彩色关键帧传播到场景的其余部分来逐帧处理视频。尽管这是一种使视频着色的简单方法，但是当场景突然发生变化时，它可能无法传播颜色。特别是，基于 RNN 的方法具有以下局限性：

1) 他们要求第一帧上色并且不能使用相关框架。
2) 它们无法在场景变化之间传播，并且因此需要精确的场景分割。 这不允许通常处理来回交替的场景在电影中完成，最终需要许多其他彩色参考。
3) 一旦出错，便会继续放大。 这个严格限制了可以传播的帧数。

与基于 RNN 的方法相比，我们的方法能够如图 2.2 所示，无缝处理多个场景或整个视频。代替使用 RNN，我们使用带有时间卷积的 CNN 和关注，这可以纳入非本地信息从多个输入帧中着色单个输出帧。

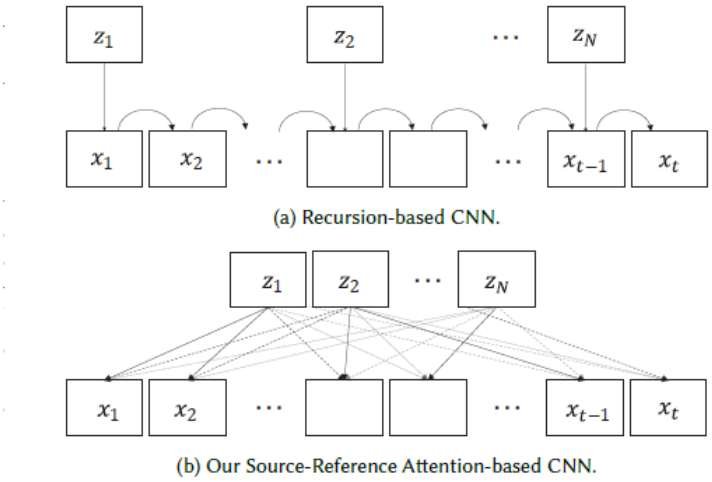

图 2.2 Comparison between recursion-based and attention-based Convolutional Neural Networks (CNN) when processing an input video x with reference images z.

基于递归的网络仅逐帧传播信息，因此无法并行处理且无法形成长期依赖关系。 每次使用新的参考图像时，都会重新开始传播，并且会丢失时间一致性。 基于源参考关注的网络（例如我们的方法）在处理任何帧时都可以使用所有参考信息。

2.3. Attention

最初开发了神经网络的注意机制用于自然语言翻译（NLT）。与人类注意力类似，对神经网络的注意力使模型专注于输入的不同部分。对于 NLT，请注意允许找到输入语言单词和之间的映射输出的语言单词，可以以不同的顺序排列。对于在自然语言处理中，已经提出了许多不同的变体，例如全球和本地注意力，正在进行大规模研究的自我注意力。

Computer vision 也已经看到了字幕的应用图像的生成，标题中的每个单词都可以专注于图像的不同部分注意。帕尔玛提出使用自我注意(self-attention) 用于图像生成，其中像素位置已明确编码。后来简化为不需要显式编码像素位置。与 我们的方法更相关的是自注意力扩展到视频分类，其中使用自注意力机制计算不同视频帧中对象的相似度。这可以改善视频的分类结果。我们的方法基于相同的概念，但是我们将其扩展为 计算输入视频帧和任意数量的参考图像之间的相似度。

3. 实验设计概念

我们的方法基于完全卷积网络，卷积神经网络的一种变体，其中仅卷积使用层。 这样可以处理图像并任何分辨率的视频。 我们采用时空混合卷积层，以及基于注意力的机制允许我们在使用过程中使用任意数量的参考彩色图像重新制作。 可以看到所提议方法的概述在图 3.1 中。

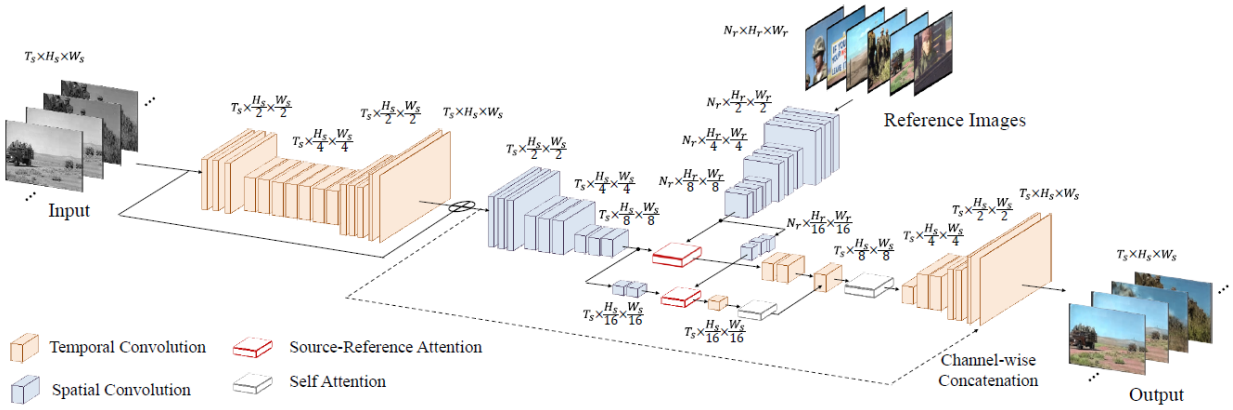

图 3.1 拟议方法概述

输入模型的黑白图像序列，并使用预处理网络对其进行恢复并用作最终输出视频的亮度通道。一种方法是，来源参考网络使用任意数量的参考彩色图像与预处理网络的输出相结合，以产生视频的最终色度通道。 引用源引用该模型在对视频着色时采用参考彩色图像中相似区域的颜色。 模型的输出是输入。

3.1. Source-Reference Attention

我们使用源引用注意来提供任意的模型可以用作的参考颜色图像数视频重播提示。特别地，源参考注意层将两个不同的可变长度体积特征图作为输入，一个对应于源数据，另一个对应于参考数据，并且允许模型利用源数据和参考数据之间的非局部相似性。这个因此，模型可以使用来自参考数据的颜色来着色源数据的类似区域。

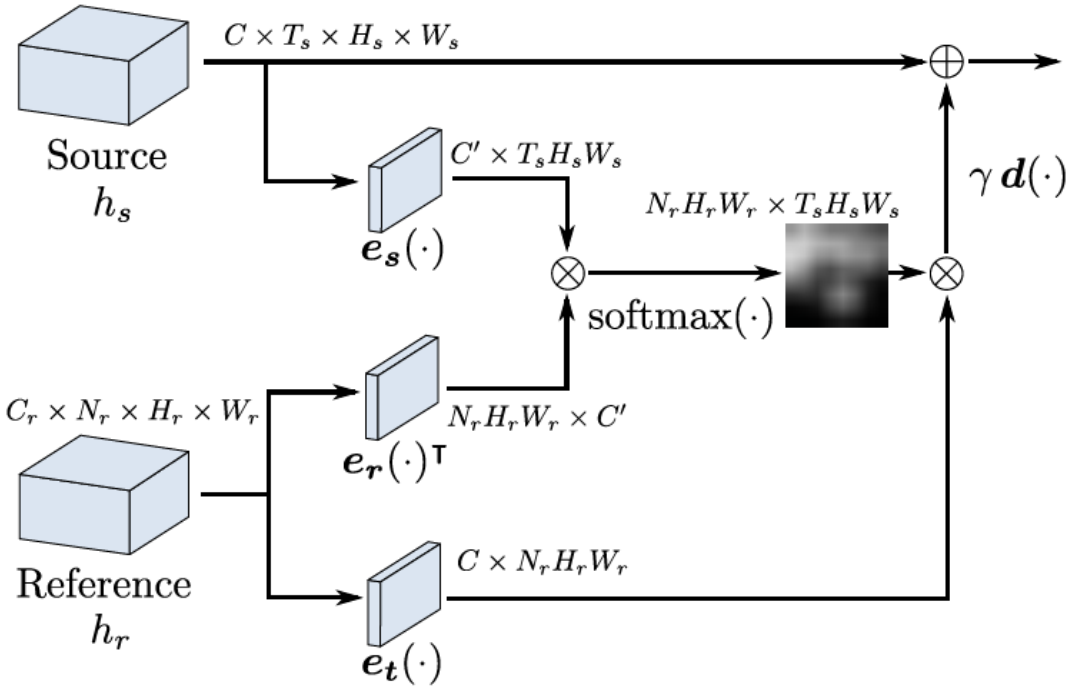

图 3.2 源代码引用注意层概述

该层将一组参考要素地图 hr 和一组源要素地图 hs 作为输入，并输出一组与源要素地图尺寸相同的新要素地图。 此注意事项允许使用参考要素中的非本地要素来执行对来源参考功能。 进行此转换，同时保留类似于纯卷积层的本地信息。 我们表示矩阵乘法表示为“⊗”，矩阵加法表示为“⊕”。 输入和输出显示了不同组件的尺寸以供参考。

3.2. Model

该模型从根本上由两个可训练的部分组成：预处理网络和来源参考网络。两者都完全以端到端的方式进行区分和培训。我们遵循完全卷积网络的最佳做法每个卷积层都由一个卷积运算符组成通过批标准化（BN）层，以及指数线性单位（ELU）激活函数，除非另有说明。除非另有说明，否则所有卷积在时域内进行空间卷积运算符使用大小为 1×3×3 的内核以及时间卷积运算符使用 3×3×3 大小的内核。所有层均使用填充 因此输出与输入的大小相同。分辨率为跨度为 1×2×2 像素的图层会减少，并增加在卷积层之前进行三线性上采样必要时。该模型的完整概述可以在图 3.1 中看到。

3.2.1. Pre-Processing Network（预处理网络）

预处理网络仅由时间卷积层组成，并在输入和输出之间使用跳过连接。 预处理网络的主要目的是从输入的灰度视频中去除伪像和噪声。 该网络使用了编码器-解码器体系结构，其中分辨率降低了一半，并通过三线性上采样最终恢复到完整大小。 表 1 显示了预处理模型体系结构的完整概述。大多数处理都是在低分辨率下进行的，以减少计算负担，并且该网络的输出用作最终输出图像的亮度通道。

Layer Type	Output Resolution	Notes
Input	$1 \times T_s \times W_s \times H_s$	Input greyscale image
TConv.	$64 \times T_s \times W_s/2 \times H_s/2$	Replication padding, spatial stride of 2
TConv. (×2)	$128 \times T_s \times W_s/2 \times H_s/2$	
TConv.	$256 \times T_s \times W_s/4 \times H_s/4$	Spatial stride of 2
TConv. (×4)	$256 \times T_s \times W_s/4 \times H_s/4$	
TConv.	$128 \times T_s \times W_s/2 \times H_s/2$	Trilinear upsampling
TConv. (×2)	$64 \times T_s \times W_s/2 \times H_s/2$	
TConv.	$16 \times T_s \times W_s \times H_s$	Trilinear upsampling
TConv.	$1 \times T_s \times W_s \times H_s$	TanH output, input is added, and finally clamped to [0, 1] range

表 1 预处理模型体系结构概述。

我们将“时间卷积”缩写为“ TConv。”。 图层不规则在注释列中指定。 当同一层连续重复几次时，我们用括号中的次数表示。

3.2.2. Source-Reference Network (源参考网络)

源参考网络形成模型的核心，并将预处理网络的输出以及任意数量的用户提供的参考彩色图像作为输入。两种形式的注意用于在计算输出色度图时允许使用非本地信息：源参考注意允许使用来自参考彩色图像的信息，从而使用户可以间接控制着色。自我注意允许使用非本地时间信息，从而增加了着色的时间一致性。对于自我注意，我们使用 sourcereference 注意层实现，并将相同的功能用于源地图和参考地图。表 2 给出了源引用模型体系结构的概述。

与预处理网络一样，该模型基于编码器-解码器体系结构，降低了分辨率以允许更有效的计算和更低的内存使用，并进行了恢复。用于最终输出。虽然时间卷积可以更好时间一致性，也使学习复杂化并增加计算负担。与预处理网络不同，源参考网络混合使用时间和空间卷积。
特别是，解码器和 1/8 中间分支使用时间卷积，而输入视频和参考的编码器图像使用空间卷积，1/16 中间分支使用两者的混合，我们发现减少了内存使用量并简化了训练，同时又不牺牲任何重新制作的准确性。此外，在参考彩色图像的情况下，由于没有必要使用图像时间卷积，因此没有时间相干性可以用。

首先，通过单独的编码器将输入视频和参考图像分为三个阶段分别减小到原始宽度和高度的 1/8。
然后将编码后的输入视频和参考视频功能分为两个分支：一个以 1/8 的宽度和高度处理视频，一个将分辨率降低到另一级，将其降低到原始宽度和高度的 1/16，以进一步处理视频。两个分支都使用源参考注意层，其他时间卷积层和自我注意层。特别是，1/16 分支在经过三线性上采样并连接到 1/8 分支输出之前，会经过自我关注层处理。
自觉地处理生成的组合特征，使其在时间上更加统一。然后，解码器使用三线性上采样在三个阶段中将特征转换为色度通道。最后，网络的输出用作图像色度，其中两个通道对应于 Lab 色彩空间的 ab 通道，而预处理网络的输出用作对应于 L 通道的图像亮度。

(a) Source and Reference Encoders.

Layer Type	Output Resolution	Notes
Input	(1 or 3) × T × W × H	3 channels (RGB) for reference, 1 channel (greyscale) for source
SConv.	64 × T × $W/2$ × $H/2$	Spatial stride of 2
SConv. (×2)	128 × T × $W/2$ × $H/2$	
SConv.	256 × T × $W/4$ × $H/4$	Spatial stride of 2
SConv. (×2)	256 × T × $W/4$ × $H/4$	
SConv.	512 × T × $W/8$ × $H/8$	Spatial stride of 2
SConv. (×2)	512 × T × $W/8$ × $H/8$	

(b) Middle 1/16 branch.

Layer Type	Output Resolution	Notes
SConv.	512 × T_s × $W_s/16$ × $H_s/16$	Input is source encoder output, spatial stride of 2
SConv.	512 × T_s × $W_s/16$ × $H_s/16$	Outputs 1/16 source
SConv.	512 × N_r × $W_r/16$ × $H_r/16$	Input is reference encoder output, spatial stride of 2
SConv. (×2)	512 × N_r × $W_r/16$ × $H_r/16$	Outputs 1/16 reference
SR Attn.	512 × T_s × $W_s/16$ × $H_s/16$	Uses 1/16 source and reference as inputs
TConv.	512 × T_s × $W_s/16$ × $H_s/16$	
Self Attn.	512 × T_s × $W_s/16$ × $H_s/16$	

(c) Middle 1/8 branch.

Layer Type	Output Resolution	Notes
SR Attn.	512 × T_s × $W_s/8$ × $H_s/8$	Input is source and reference encoder output
TConv. (×2)	512 × T_s × $W_s/4$ × $H_s/4$	
TConv. (×2)	512 × T_s × $W_s/4$ × $H_s/4$	Output of the 1/16 branch is concatenated to the input
Self Attn.	512 × T_s × $W_s/4$ × $H_s/4$	

(d) Decoder.

Layer Type	Output Resolution	Notes
TConv.	256 × T_s × $W_s/8$ × $H_s/8$	
TConv.	128 × T_s × $W_s/4$ × $H_s/4$	Trilinear upsampling
TConv.	64 × T_s × $W_s/4$ × $H_s/4$	
TConv.	32 × T_s × $W_s/2$ × $H_s/2$	Trilinear upsampling
TConv.	16 × T_s × $W_s/2$ × $H_s/2$	
TConv.	8 × T_s × W_s × H_s	Trilinear upsampling
TConv.	2 × T_s × W_s × H_s	Sigmoid output represents chrominance

表 2 源参考模型体系结构概述

该模型将预处理模型的输出和一组参考图像作为输入。 这两个输入均由单独的编码器（a）处理，然后在对应于 1/16 宽度和高度（b）和 1/8 宽度和高度（c）的两个不同的中间分支中进行处理，然后再解码为的色度通道 使用解码器（d）输出视频。 我们将空间卷积缩写为“ SConv。”，将时间卷积缩写为“ TConv。”，并将“源引用注意”缩写为“ SR Attn”。 对于源编码器和参考编码器，我们通常将时间维称为 T，其中 T = Tr 表示参考编码器，T = Ts 表示源编码器。 我们在注释列中指定图层不规则性。 当同一层连续重复几次时，我们用括号中的次数表示。

3.3. TRAINING

我们使用人工策划的监督训练数据训练模型。为了提高结果的一般性和质量，我们执行了大量的综合数据扩充和基于实例的薄膜劣化。

3.3.1. Objective Function (目标功能)

我们以两个 L1 损失的线性组合以完全监督的方式训练模型。
特别地，我们使用由 Lab 色彩空间和参考彩色图像 z 组成的，由退化的黑白视频 x 和恢复的彩色视频对组成的监督数据集 D，这些彩色视频分为亮度 yl 和色度 yab，并参考彩色图像 z，并优化以下表达式：

$$\arg\min_{\theta,\phi} \mathbb{E}_{(x,y_l,y_{ab},z)\in\mathcal{D}} \|P(x;\theta) - y_l\| + \beta \|S(P(x;\theta), z;\phi) - y_{ab}\|,$$

Eq（3）

其中 P 是权重为 θ 的预处理模型，S 是权重为 ϕ 的源参考模型， β ∈R 是权重超参数。

使用一批视频（每批视频具有 5 个连续帧）进行训练，这些视频是从训练数据中随机选择的。
对于每个 5 帧视频，从[0，6]范围内均匀选择随机数量的颜色参考图像 z。 如果参考的数量不为 0，则从输入帧的五个相邻帧中选择参考图像之一，而其余的参考图像则从整个训练数据集中随机采样。

3.3.2. Training Data

我们的数据集基于 YouTube-8M 数据集，该数据集包含大约 800 万个视频，对应于大约 50 万小时的视频数据。 该数据集带有 4,803 个我们未使用的视觉实体的注释。 我们将视频转换为黑白图像并对其进行破坏，以模拟旧电影的退化，从而为我们的模型创建监督的训练数据。

由于 YouTube-8M 数据集大部分是自动创建的，因此大量视频描述了游戏玩法，黑白视频，固定摄像机的静态场景以及不自然着色的场景，例如带有现场音乐的俱乐部。 我们从完整的数据集中随机选择视频，并对其进行手动注释，以适合训练和评估重新制作模型。 特别是，我们最终获得了 1,569 个视频，总计 10,243,010 帧，其中我们使用 1,219（7,993,132 帧）来训练模型，使用 50（321,306）进行验证，并使用 300（1,928,572）进行测试。

3.3.3. Data Augmentation

我们对输入视频进行大量数据扩充，地面实况视频和参考图像具有两个目标：首先，我们希望增加模型对不同类型视频的通用性；其次，我们希望模型能够恢复输入中常见的不同伪像视频，例如模糊或低对比度。与基于示例的恶化同时进行此数据增强，这进一步恶化了输入灰度视频。

我们使用一批 5 帧视频及其相关的参考图像，分辨率为 256×256 像素。作为数据扩充，我们执行了大量的转换，这些转换一起影响输入视频 x 和地面真实视频 y =（yl，yab），仅影响输入视频 x，仅影响参考图像 z 或前三个图像的任意组合。表 3 显示了我们应用的不同转换的概述，其中包括亮度，对比度，JPEG 噪声，高斯噪声，模糊和饱和度的变化。

Name	Target	Prob.	Range	Notes
Horiz. Flip	$(x, y), z$	50%	-	
Scaling	(x, y)	100%	$\mathcal{U}(256, 400)$	Size of the smallest edge (px), randomly crops
Rotation	(x, y)	100%	$\mathcal{U}(-5, 5)$	In degrees
Brightness	(x, y)	20%	$\mathcal{U}(0.8, 1.2)$	
Contrast	(x, y)	20%	$\mathcal{U}(0.9, 1.0)$	
JPEG	x, z	90%	$\mathcal{U}(15, 40)$	Encoding quality
Noise	x, z	10%	$\mathcal{N}(0, 0.04)$	Gaussian
Blur	x	50%	$\mathcal{U}(2, 4)$	Bicubic down-sampling
Contrast	x	33%	$\mathcal{U}(0.6, 1.0)$	
Scaling	z	100%	$\mathcal{U}(256, 320)$	Size of the smallest edge (px), randomly crops
Saturation	z	10%	$\mathcal{U}(0.3, 1.0)$	

表 3 使用的不同类型的数据扩充概述
培训期间。

目标是指正在扩充的数据。 括号中的值表示对两个变量共同执行相同的转换，而不是独立进行。 概率表示特定转换可能发生的可能性，范围是转换参数的采样方式。 我们注意到，在输入视频 x 和目标视频 y 的情况下，对视频中的所有帧进行相同的变换，而在参考图像 z 的情况下，对每个图像按原样独立进行变换 彼此无关。

3.3.4. Example-based Deterioration

除了所有不同的数据增强技术之外，我们还从 6152 个示例数据集中模拟了胶片介质的劣化情况，例如分形噪声，颗粒噪声，灰尘和划痕。这些恶化示例是通过使用关键词“胶片噪声”的网络搜索手动收集的，并且还使用 Adobe

After Effects 等软件生成的。对于产生的噪声，使用分形噪声来生成基本噪声模式，然后可以通过修改对比度，亮度和色调曲线来获得划痕和类似灰尘的噪声，从而对其进行改进。总共下载了 4,340 张噪声图像，并生成了 1,812 张。图 3.3 中显示了一些劣化示例。尤其是，由于这些劣化效果模拟了支撑胶片的物理介质的劣化，因此它们被实现为加性噪声：噪声数据被随机添加到输入的灰度视频中，每个帧独立。此外，它们彼此独立地添加并组合以创建增强的输入视频。

对于所有噪声，我们使用与输入视频类似的数据增强技术。 尤其是，噪声图像会随机缩放，以使最短边缘在[256，720]像素之间，以 50％的概率水平和垂直翻转，在[-5，5]度之间随机旋转，裁剪为 256×256 像素 ，按 U（0.5，1.5）随机缩放，并随机减去或添加到原始图像。 一些生成的训练示例如图 3.4 所示。

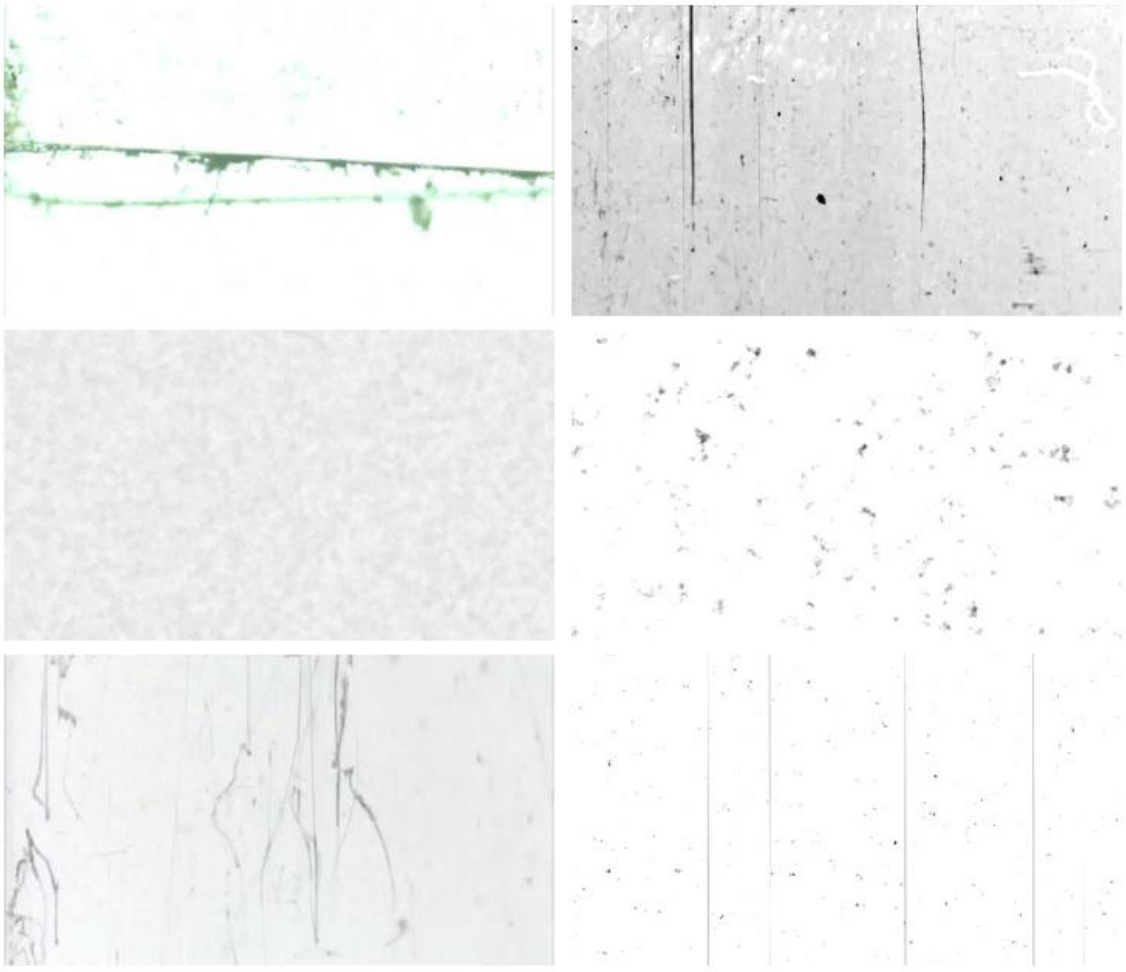

图 3.3 基于示例的恶化效果。

这些效果是脱机生成的，并存储为图像数据集，然后可以将其作为附加噪声应用于训练数据输入。

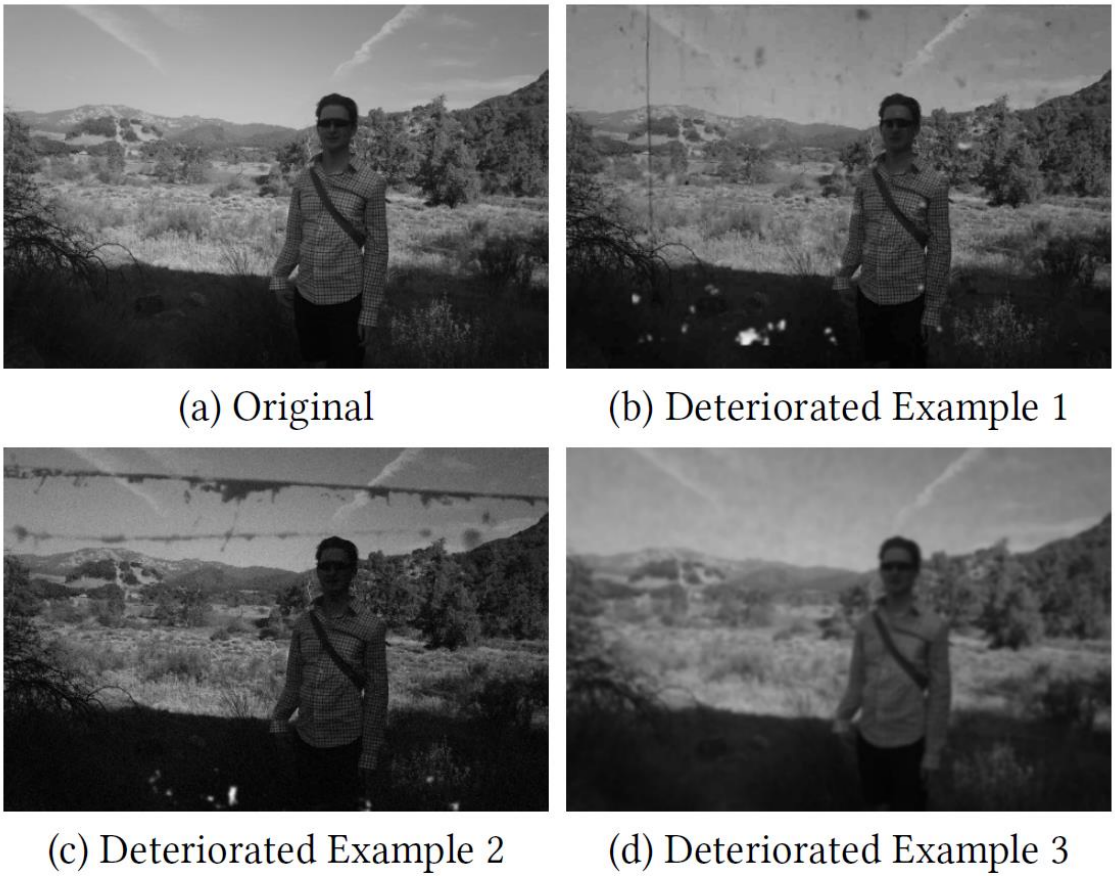

图 3.4 应用于黑白图像的合成劣化效果的示例

（a）对于原始图像，随机添加（b-d）各种类型的基于算法和基于示例的劣化效果，例如 JPEG 压缩伪影和胶片划痕。 在公共领域获得许可的视频。

3.3.5. Optimization

最初分别对预处理网络和源参考网络进行了 500,000 次迭代的培训。 然后，通过优化等式（3）以端到端的方式一起训练他们。 对于优化方法，我们依赖于 ADADELTA 算法，该算法是随机梯度下降的一种变体，可通过启发式方法估算学习速率参数，因此无需进行超参数调整。

4. 实验结果与分析比较

我们在 γ = 10−4 且批次大小为 20 的数据集上训练模型。我们使用验证损失最小的模型作为最终模型。 我们评估定量和定性并与现有方法进行比较。

4.1. 与现有方法的比较 (Comparison with Existing Approaches)

我们比较了我们的方法与现有的方法和强大的基线定量评价的结果。具体而言，对于复原，我们与[Zhang et al.2017b]和[Yu et al.2018]的方法进行比较，对于彩色化，我们与[Vondrick et al.2018]的基于传播的方法和[Zhang et al.2017a]的单图像交互方法进行比较。对于两种修复方法，即关节修复和着色，我们将其与所有可能的修复和着色方法组合进行比较，例如[Zhang et al.2017b]和[Vondricket al.2018]结合使用。[Zhang et al.2017b] 和 [Yu et al.2018] 的方法包括用于单个图像恢复的深度残差卷积神经网络。我们注意到， [Yu et al.2018] 的方法是 [Fan et al.2018] 的延伸，是 NTIRE 2018 超分辨率图像挑战赛的冠军。我们修改了[Yu et al.2018] 的模型，在最后移除上采样层，因为目标任务是恢复而不是超分辨率。

我们使用由 300 个来自 Youtube-8M 数据集的视频组成的测试集进行比较。对于每一个视频，我们随机抽取 90 或 300 帧的子集，并使用该子集作为基本事实。鉴于这些视频既没有噪声也没有退化，我们采用相同的方法生成训练数据，以生成退化的输入进行评估。对于基于实例的退化效果，我们使用不同于训练集的图像集来评估泛化效果。我们使用峰值信噪比（PSNR）作为评价指标，在恢复任务中仅使用亮度通道，在着色任务中仅使用色度通道，在恢复任务中使用所有图像通道来计算 PSNR。

对于参考彩色图像，在 90 帧子集的情况下，我们仅提供第一帧作为参考图像，而在 300 帧子集的情况下，我们提供从第一帧开始的每 60 帧作为参考图像。对于我们的方法，所有的参考框架都是在任何时候提供的。

在[Vondrick et al.2018] 的方法的情况下，由于其仅传播颜色并且不能自然地处理多个参考图像，因此我们在必要时用新的参考图像替换输出图像，如图 2.2 所示。我们注意到，所有方法都使用了所有视频的相同随机子集。

4.1.1. Remastering Results

由于没有一种方法可以处理视频的重拍，我们将其与管道方法进行比较，即首先使用[Zhang et al.2017b]或[Yu et al.2018]的方法处理视频，然后使用[Vondrick et al.2018]或[Zhang et al.2017a]的方法在输出上传播参考颜色。我们还提供了一个基线的结果，该基线由我们的完整方法组成，无需联合训练，即恢复和着色网络是独立训练的。

结果见表 4。在基于管道的方法中，我们发现，虽然它们具有相似的性能，但是[Zhang 等人 2017b]和[Zhang 等人 2017a]的组合给出了最高的性能。然而，我们的方法优于现有的基于管道的方法和不使用联合训练的强基线。这表明，尽管修复和着色模型在联合训练之前是先独立训练的，但联合训练对提高最终结果的质量起着重要作用。有趣的是，当视频越长，参考彩色图像越多，现有方法的性能越差时，我们的方法的性能也会提高。这可能是因为所有的参考颜色图像都被用来重制每一帧。图 4.1 中显示了几个随机选择的示例，其中我们可以看到现有的方法不能同时去除噪声和传播颜色，而我们的方法在这两种情况下都表现良好。

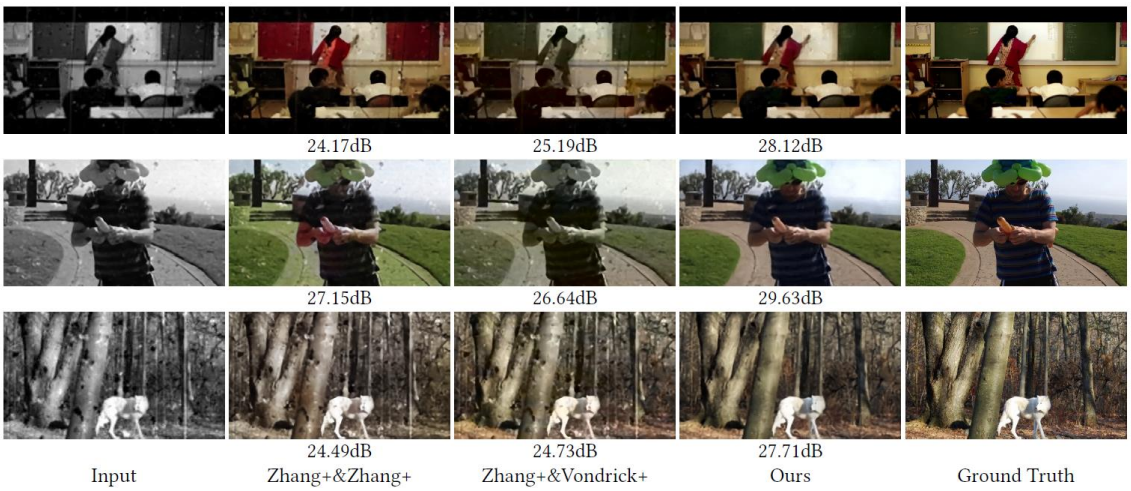

图 4.1 从 Youtube-8M 测试数据集中随机抽取示例，并带有降级噪声。

我们展示了几个例子中的一个框架，并将我们的方法与[Zhang et al.2017b]和[Zhang et al.2017a]以及[Zhang et al.2017b]和[Vondrick et al.2018]的组合方法进行了比较。第一列显示已被噪声恶化的输入帧，下两列对应两种方法的重拍结果，最后一列显示地面实况视频。每种方法的 PSNR 都显示在每个图像下面。

Approach	Frames	# Ref.	PSNR
Zhang+[2017b]&Zhang+[2017a]	90	1	27.13
	300	5	27.31
Yu+[2018]&Zhang+[2017a]	90	1	26.43
	300	5	26.59
Zhang+[2017b]&Vondrick+[2018]	90	1	26.43
	300	5	26.60
Yu+[2018]&Vondrick+[2018]	90	1	26.85
	300	5	26.89
Ours w/o joint training	90	1	29.07
	300	5	29.23
Ours	90	1	**30.83**
	300	5	**31.14**

表 4 定量修复结果

我们将我们模型的结果与[Zhang et al.2017b]的方法恢复每一帧的结果进行比较，并将参考颜色与[Vondrick et al.2018]的方法在 Youtube-8M 数据集中的综合退化视频上传播，并与不使用联合训练的由我们模型组成的基线进行比较。我们进行了两种类型的实验：一种是使用一个参考帧的随机 90 帧子集，另一种是使用 5 个参考帧的随机 300 帧子集。

4.1.2. Restoration Results

我们将我们的方法与[Zhang et al.2017b]，[Yu et al.2018]的方法以及视频恢复的基线进行了比较。

基线由我们的预处理模型组成，没有将输入添加到输出的跳过连接。由于不添加颜色，因此不提供参考颜色图像，并且仅使用 300 帧子集进行评估。结果见表 5。我们可以看到，基线 [Zhang et al.2017b] 的方法和 [Yu et al.2018] 的方法表现相似，而我们的完整预处理模型（带跳过连接）的性能都优于这两种方法。示例结果如图 4.2 所示。

图 4.2 带有降级噪声的 Youtube-8M 测试数据集的恢复结果

我们展示了几个例子中的一个框架，并将我们的方法与[Zhang et al.2017b]和[Yu et al.2018]的方法进行了比较。第一列显示已被噪声恶化的输入帧，下三列对应每种方法的黑白恢复，最后一列对应地面实况视频。

Approach	Frames	# Ref.	PSNR
[Zhang et al. 2017b]	300	-	25.08
[Yu et al. 2018]	300	-	24.49
Ours w/o skip connection	300	-	24.73
Ours	300	-	**26.13**

表 5 定量恢复结果

我们将预处理网络的结果与[Zhang et al.2017b]、[Yu et al.2018]的方法进行了比较，并将不使用跳过连接的方法的基线用于从 Youtube-8M 数据集中恢复综合恶化的视频。

4.1.3. Colorization Results

我们将[Zhang et al.2017a]使用全局提示的方法、[Vondrick et al.2018]的方法和两条基线进行了比较：一条基线由没有时间卷积的源参考网络组成，另一条基线没有用于着色的自我注意。结果如表 6 所示，我们可以看到我们的方法优于现有的方法和基线。

与 remastering 的情况类似，我们的方法在具有附加参考图像的较长视频上的性能明显更好，这表明源参考注意的能力：不仅可以对具有许多参考图像的长序列着色，而且有利于性能。一个有趣的结果是，自我关注在我们的模型中起着关键作用。我们认为这是因为它允许使用来自整个图像的信息来计算每个

输出像素，如果不使用自我注意，这将需要更多的卷积层。示例结果如图 4.3 所示。

图 4.3 Youtube-8M 测试数据集上的着色结果。

我们展示了几个示例中的一帧，并将我们的方法与[Zhang et al.2017a]的不使用参考图像的彩色化方法以及基于 RNN 的使用参考图像的方法[Vondrick et al.2018]进行了比较。第一列显示输入帧，接下来的三列对应于每种方法的彩色化，最后一列对应于参考图像。注意，输入帧与参考图像不是同一帧。

Approach	Frames	# Ref.	PSNR
[Zhang et al. 2017a]	90	1	31.28
	300	5	31.16
[Vondrick et al. 2018]	90	1	31.55
	300	5	31.70
Ours w/o temporal conv.	90	1	28.46
	300	5	28.51
Ours w/o self-attention	90	1	29.00
	300	5	28.72
Ours	90	1	**34.94**
	300	5	**36.26**

表 6 定量着色结果

我们将源参考网络的着色结果与[Zhang et al.2017a]使用全局提示的方法和[Vondrick et al.2018]使用 Youtube-8M 数据集的视频着色方法进行了比较。我们进行了两种类型的实验：一种是使用一个参考帧的随机 90 帧子集，另一种是使用 5 个参考帧的随机 300 帧子集。

4.2. 定性结果

我们在图 4.4 中展示了各种具有挑战性的真实世界复古电影示例的定性结果。由于视频最初是彩色的，因此我们使用来自原始视频的图像作为参考图像，然后将我们的重影方法和管道去噪方法与[Zhang et al.2017b]的方法进行比较，然后使用[Vondrick et al.2018]的方法添加颜色。我们可以看到我们的方法是如何能够执行一致的重排，而现有的方法失去了跟踪着色，并不能产生令人满意的结果，这与我们的定量评估是一致的。我们还使用[Zhang et al.2017b] 的方法对图 4.5 中的复古胶片的修复结果进行了定性比较。我们可以看到 [Zhang et al.2017b] 的方法如何恢复小噪声，但在较大噪声下失败。我们的方法既能处理小噪声又能处理大噪声，同时还能锐化输入图像。

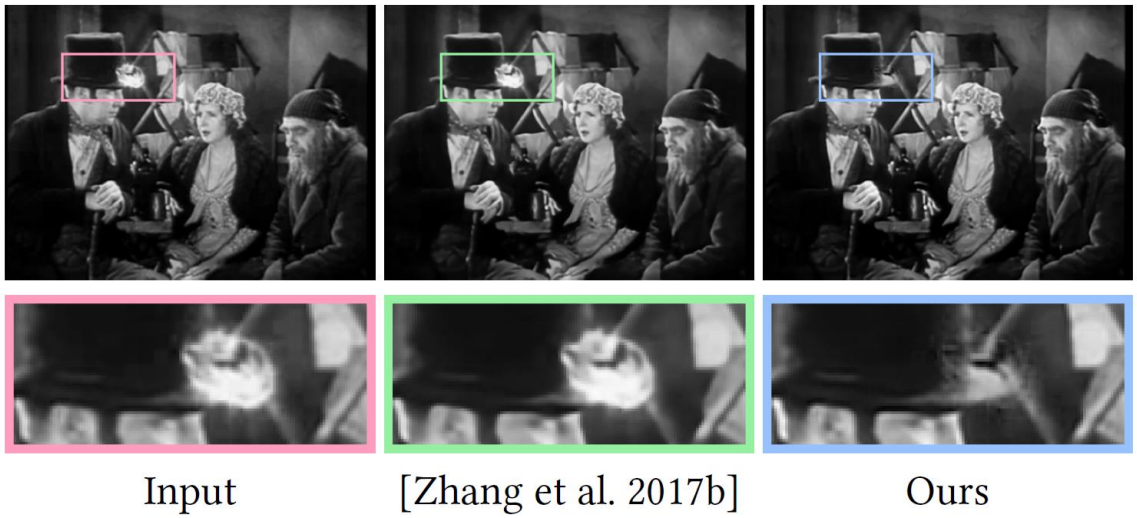

图 4.5 复古胶片上的恢复结果。

我们与[Zhang et al.2017b]的方法进行了比较，并在最下面一行显示了放大的方框区域。我们可以看到，相对较大的噪音是“inpainted”与我们的网络。

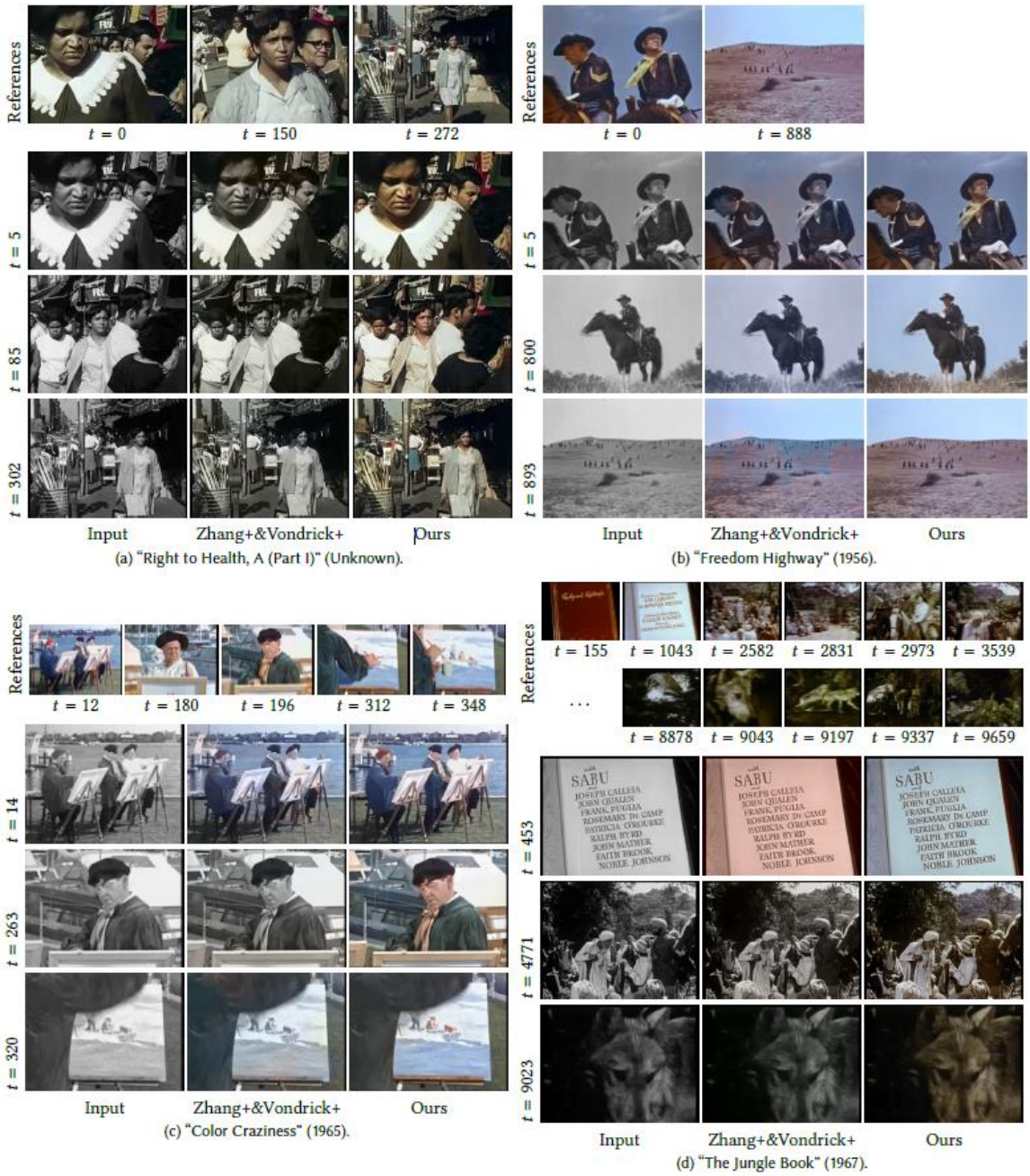

图 4.4 与 Zhang+[2017b]和 Vondrick+[2018]联合方法的定性比较。
我们在第一行显示参考彩色图像及其时间戳。然后显示从输入视频和输出视频中提取的四个不同帧。注意，（d）的例子是用 41 个参考图像重新合成的，我们只显示其中的一个子集。

4.3. 计算时间

对于 528×400 像素的输入视频，我们的方法使用 Nvidia GTX 1080Ti GPU 每帧花费 69ms，其中 4ms 对应于恢复阶段，而 65ms 对应于着色阶段。

5. 实验总结

我们提出了一种基于时间卷积网络的复古胶片重制方法，该方法具有源参考注意机制，允许使用任意数量的参考彩色图像。尽管源参考注意机制是将参考图像合并到处理框架中的一个强大工具，并且可以处理任何分辨率的视频，但是它会受到 O（NrHrWrTsHsWs）内存使用的影响。因此，可用的系统内存将限制可处理的最大分辨率。然而，在实践中，由于电影技术的限制，大多数老电影都是以低分辨率存储的，因此它们不必以基于注意力的机制无法实现的分辨率进行处理。

目前，所提出的方法依赖于完全监督学习，并且不能填充丢失的帧，也不能在如图 5.1 所示的许多帧期间导致图像的大区域丢失的极端退化。在这些情况下，有太多的信息丢失，这使得它不可能重拍，他们将需要基于图像完成的方法来重拍新的可信部分的视频，这是超出了这项工作的范围。

我们的模型具有 15 帧的时间分辨率，在大多数视频中大约相当于半秒，这会导致输出视频的时间一致性很小。作为参考，现有方法使用较小的量，例如 4 帧[Vondrick et al.2018]或 10 帧[Lai et al.2018]。虽然可以提高时间分辨率，但这会导致收敛速度和计算速度变慢。虽然盲视频时间一致性技术可以缓解这一问题[Bonneel et al.2015；Lai et al.2018] ，但我们发现，尽管盲视频时间一致性技术能够略微改善时间一致性，但其代价是结果显著恶化。我们相信，将这种方法与我们的模型和端到端训练相结合是一种在不牺牲结果质量的情况下提高时间一致性的可能方法。

我们注意到，尽管这项工作在复古电影的进展，由于任务的复杂性，它仍然是一个开放的问题，在计算机图形学。与迄今为止的大多数图像和视频研究不同，复古电影提出了一个更加困难和现实的问题，如图 2.1 所示，我们希望这项工作能够进一步促进这一主题的研究。

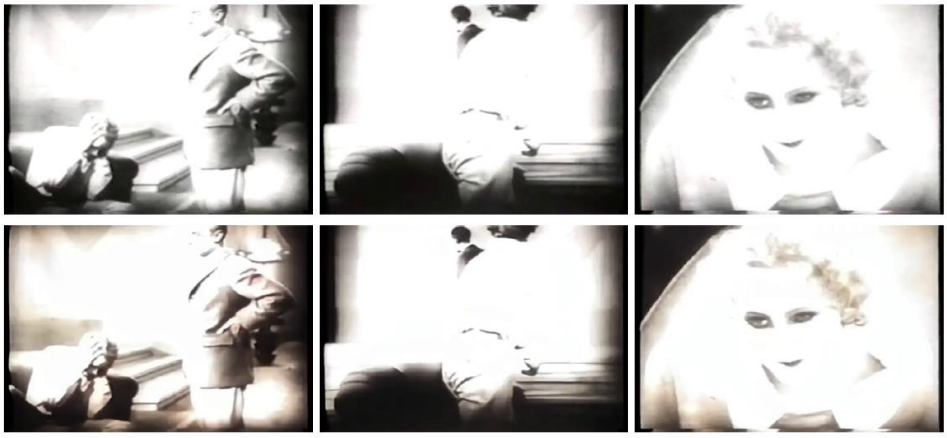

图 5.1　我们方法的局限性

严重劣化的薄膜的例子，用目前的方法不可能重制。第一行显示原始输入视频的帧，第二行显示我们方法的输出。

References

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Apostol (Paul) Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. 2016. YouTube-8M: A Large-Scale Video Classification Benchmark. In *arXiv:1609.08675*. https://arxiv.org/pdf/1609.08675v1.pdf

Xiaobo An and Fabio Pellacini. 2008. AppProp: All-pairs Appearance-space Edit Propagation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 27, 3 (Aug. 2008), 40:1–40:9.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. (2015).

Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novak, Alex Harvill, Pradeep Sen, Tony Derose, and Fabrice Rousselle. 2017. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 36, 4 (2017), 97–1.

Nicolas Bonneel, James Tompkin, Kalyan Sunkavalli, Deqing Sun, Sylvain Paris, and Hanspeter Pfister. 2015. Blind Video Temporal Consistency. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 34, 6 (2015).

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive Exploration of Neural Machine Translation Architectures. In *Conference on Empirical Methods in Natural Language Processing*.

Chakravarty R Alla Chaitanya, Anton S Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. 2017. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 36, 4 (2017), 98.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Conference on Empirical Methods in Natural Language Processing*.

Alex Yong-Sang Chia, Shaojie Zhuo, Raj Kumar Gupta, Yu-Wing Tai, Siu-Yeung Cho, Ping Tan, and Stephen Lin. 2011. Semantic Colorization with Internet Images. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 30, 6 (2011), 156:1–156:8.

Djork-Arne Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). In *International Conference on Learning Representations*.

K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. 2007. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing* 16, 8 (2007), 2080–2095.

Xiaopei Liu, Liang Wan, Yingge Qu, Tien-Tsin Wong, Stephen Lin, Chi-Sing Leung, and Pheng-Ann Heng. 2008. Intrinsic Colorization. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 27, 5 (December 2008), 152:1–152:9.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Conference on Empirical Methods in Natural Language Processing*.

M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian. 2012. Video Denoising, Deblocking, and Enhancement Through Separable 4-D Nonlocal Spatiotemporal Transforms. *IEEE Transactions on Image Processing* 21, 9 (2012), 3952–3966.

M. Maggioni, E. Sanchez-Monge, and A. Foi. 2014. Joint Removal of Random and Fixed-Pattern Noise Through Spatiotemporal Video Filtering. *IEEE Transactions on Image Processing* 23, 10 (2014), 4282–4296.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. 2001. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *International Conference on Computer Vision*.

Simone Meyer, Victor Cornillere, Abdelaziz Djelouah, Christopher Schroers, and Markus Gross. 2018. Deep Video Color Propagation. In *British Machine Vision Conference*.

Ankur P Parikh, Oscar Tackstrom, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Conference on Empirical Methods in Natural Language Processing*.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, and Alexander Ku. 2018. Image Transformer. In *International Conference on Machine Learning*.

Francois Pitie, Anil C. Kokaram, and Rozenn Dahyot. 2007. Automated Colour Grading Using Colour Distribution Transfer. *Computer Vision and Image Understanding* 107, 1-2 (July 2007), 123–137.

Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. 2001. Color Transfer between Images. *IEEE Computer Graphics and Applications* 21, 5 (sep 2001), 34–41.

Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. 2017. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Yu-Wing Tai, Jiaya Jia, and Chi-Keung Tang. 2005. Local Color Transfer via Probabilistic

Segmentation by Expectation-Maximization. In *IEEE Conference on Computer Vision and Pattern Recognition*. 747–754.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Conference on Neural Information Processing Systems*.

Thijs Vogels, Fabrice Rousselle, Brian McWilliams, Gerhard Rothlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novak. 2018. Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 37, 4 (2018), 124.

Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. 2018. Tracking emerges by colorizing videos. In *European Conference on Computer Vision*.

XiaolongWang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. 2002. Transferring Color to Greyscale Images. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 21, 3 (July 2002), 277–280.

Fuzhang Wu, Weiming Dong, Yan Kong, Xing Mei, Jean-Claude Paul, and Xiaopeng Zhang. 2013. Content-Based Colour Transfer. 32, 1 (2013), 190–203.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*.

Li Xu, Qiong Yan, and Jiaya Jia. 2013. A Sparse Control Model for Image and Video Editing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 32, 6 (Nov. 2013), 197:1–197:10.

Jiahui Yu, Yuchen Fan, Jianchao Yang, Ning Xu, Zhaowen Wang, Xinchao Wang, and Thomas S. Huang. 2018. Wide Activation for Efficient and Accurate Image Super-Resolution. *CoRR* abs/1808.08718 (2018). arXiv:1808.08718

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR* abs/1212.5701 (2012).

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2018a. Self-Attention Generative Adversarial Networks. *arXiv preprint arXiv:1805.08318* (2018).

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. 2017b. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing* 26, 7 (2017), 3142–3155.

Kai Zhang, Wangmeng Zuo, and Lei Zhang. 2018b. FFDNet: Toward a Fast and Flexible Solution for CNN based Image Denoising. *IEEE Transactions on Image Processing* (2018).

Richard Zhang, Phillip Isola, and Alexei A Efros. 2016. Colorful image colorization. In *European Conference on Computer Vision*.

Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. 2017a. Real-Time User-Guided Image Colorization with Learned Deep Priors. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 9, 4 (2017).