© 2023年 IEEE。允许个人使用本材料。所有其他用途，包括在任何当前或未来媒体中重印或再版本材料以用于广告或宣传目的、创作新的汇编作品、转售或再分发至服务器或列表，或在其他作品中重复使用本作品的任何受版权保护部分，均须事先获得 IEEE 的许可。数字对象标识符 10.1109/TIP.2023.3298537

arXiv:2303.11591v2 [cs.CV] 4 Aug 2023

SVCNet：采用时间聚合的基于涂鸦的视频着色网络

Yuzhi Zhao，*IEEE* 研究生会员，Lai-Man Po，*IEEE* 高级会员，Kangcheng Liu，*IEEE* 会员，Xuehui Wang，*IEEE* 学生会员，Wing-Yin Yu，*IEEE* 学生会员，Pengfei Xian，*IEEE* 学生会员，Yujia Zhang、Mengyang Liu

摘要——**本文提出一种采用时间聚合的基于涂鸦的视频着色网络，称为 SVCNet。该网络能够根据用户给定的不同颜色涂鸦为单色视频着色。它解决了基于涂鸦的视频着色领域中的三个常见问题：着色鲜艳度、时间一致性和颜色溢出。为提升着色质量并增强时间一致性，我们在 SVCNet 中采用两个依次运行的子网络，分别用于精确着色和时间平滑。第一阶段包含一个金字塔特征编码器，用于将颜色涂鸦与灰度帧结合；还包含一个语义特征编码器，用于提取语义信息。第二阶段通过聚合相邻着色帧（作为短程连接）和第一帧着色帧（作为长程连接）的信息，对第一阶段的输出进行微调。为减轻颜色渗漏伪影，我们同时进行视频着色和分割。此外，我们将大多数操作设置在固定的小图像分辨率下，并在 SVCNet 尾部使用超分辨率模块来恢复原始尺寸。这使 SVCNet 能够在推理时适应不同的图像分辨率。最后，我们在 DAVIS 和 Videvo 基准数据集上评估了所提出的 SVCNet。实验结果表明，与其他知名的视频着色方法相比，SVCNet 能够生成质量更高、时间一致性更好的视频。代码和模型可在 https://github.com/zhaoyuzhi/SVCNet 获取。**

索引词—**视频着色、基于涂鸦的着色、时间聚合、分割。**

I. 引言

视频着色是为单色视频赋予合理色彩的过程。受成像技术限制，许多老电影以黑白格式保存。人们非常希望观看彩色视频。近年来，深度神经网络在视频复原和着色领域都取得了巨大进展。因此，利用深度神经网络复原逼真且色彩丰富的视频已成为可能。

视频着色的主要难点在于连续帧的着色鲜艳度和时间一致性。此外，颜色溢出（即颜色扩散到物体边界之外）也是一项挑战。现有方法有许多解决这些问题的方案，通常可归为以下四类：

1）图像着色与时间平滑 [1]–[4]；2）图像着色与颜色传播 [5]–[8]；3）全自动视频着色 [9]–[11]；4）基于样例的视频着色 [12]–[14]。

前三类方法不依赖示例图像和彩色涂鸦等额外引导。为了学习灰度到彩色的映射，它们通常采用 ImageNet 等大型训练集 [15]，以学习丰富的数据先验。三类方法之间的差异显而易见。由于类别 1）依赖预训练图像着色方法，且仅对单幅图像着色结果进行微调，因此无法保证视频连续性。类别 2）与类别 1）类似，但只使用第一帧着色帧。它过度依赖长程连接，因而容易忽略每一帧自身的特征。类别 3）联合学习着色与时间平滑，因此表现优于类别 1）和 2）。然而，由于与彩色格式（例如，*RGB*、*YUV*和*CIE Lab*）相比，灰度格式会损失过多信息，因此这类方法难以预测真实的颜色嵌入。此外，它们可能会因时间约束而牺牲色彩丰富度。为提高着色质量，类别 4）引入示例图像来引导视频着色。尽管它生成的视频比其他方法色彩更丰富，但需要一张相对准确、与单色输入的彩色版本相似的示例图像。

为获得优于现有方法的视频着色质量并尽量减少颜色渗漏伪影，我们提出了首个基于涂鸦的视频着色框架 SVCNet，其数据流如图 1 所示。与以往方案相比，我们作出了四项改进：

1）提升着色鲜艳度：SVCNet 包含两个阶段，分别用于精确着色（CPNet）和时间平滑（SSNet）。CP Net 采用多输入多输出架构，可获得更好的着色质量；

2）增强时间一致性：SVCNet 同时采用双向投影和长程连接，有效聚合视频的时间信息；

3）减少颜色渗漏伪影：SVCNet 在执行视频着色的同时，还学习一项额外的辅助分割任务，以尽量减少颜色渗漏伪影；

4）减少人工工作：SVCNet 只需要稀疏的颜色涂鸦，而不需要精确的样例图像。

稿件收到日期：2022年11月14日，修订日期：2023年3月21日，接收日期：2023年7月20日。（通讯作者：*Yuzhi Zhao*。）Y. Zhao、L.-M. Po、W.-Y. Yu 和 P. Xian 任职于中国香港香港城市大学电子工程系（电子邮箱：yzzhao2-c@my.cityu.edu.hk；eelmpo@cityu.edu.hk；wingyinyu8-c@my.cityu.edu.hk；xian.pf@my.cityu.edu.hk）。K. Liu 任职于新加坡南洋理工大学计算机科学与工程学院（电子邮箱：kangcheng.liu@ntu.edu.sg）。X. Wang 任职于中国上海上海交通大学人工智能研究院（电子邮箱：wangxuehui@sjtu.edu.cn）。Y. Zhang 和 M. Liu 任职于中国深圳腾讯控股有限公司旗下腾讯视频（电子邮箱：yujiazhang@tencent.com；myleonliu@tencent.com）。

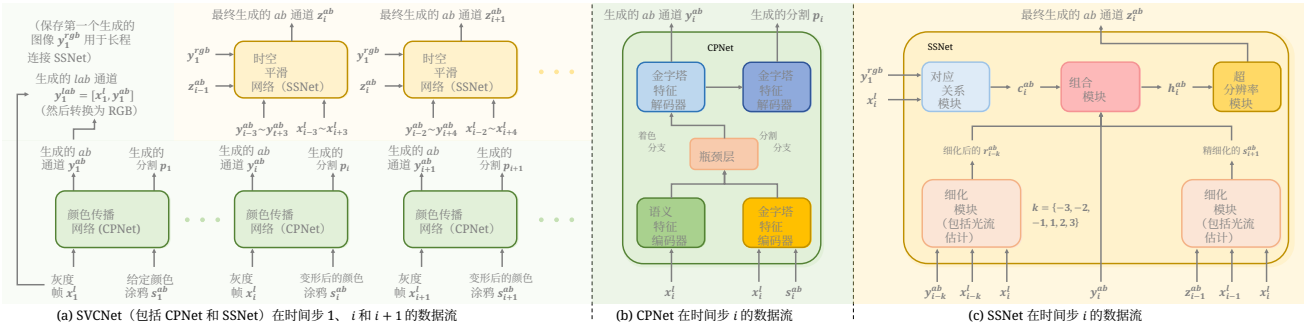


图1。（a）SVCNet（包括CPNet和SSNet）、（b）CPNet和（c）SSNet的数据流示意图。用户只需为第一帧 s_1^{ab}提供颜色涂鸦。后续的颜色涂鸦 s_2^{ab}-s_t^{ab}通过使用前向光流对 s_1^{ab}进行变形获得。

SVCNet 包含两个子网络：颜色传播子网络（CPNet）和时空平滑子网络（SSNet）。其数据流如图 1 所示。首先，CPNet 是一种多输入多输出架构，如图 1（b）所示。它包含两个编码器，用于从灰度输入中提取语义信息并融合两个输入的信息，从而实现精确的涂鸦颜色传播，并确保即使没有颜色涂鸦时也能获得良好的着色质量。此外，它还包含两个解码器分支，其中着色分支输出颜色嵌入，另一个分支输出对应的分割图。训练时，分割分支通过反向传播辅助着色分支。它促使网络划分出清晰的边界，从而帮助 CPNet 减少颜色渗漏伪影。其次，SSNet 会细化 CPNet 生成的每一帧着色帧，如图 1（c）所示。一方面，它通过细化模块，基于先前帧、当前帧和后续帧执行双向投影，形成短程连接。另一方面，它通过对应模块提取并对齐第一帧着色帧，将其作为长程连接。之后，组合模块会聚合所有信息。因此，SSNet 能够以令人满意的时间一致性为视频着色。我们注意到，颜色嵌入（即*CIE Lab*颜色空间中的*ab* 通道）比边缘稀疏得多。基于这一点，并受到 [16]–[19]，的启发，我们将上述所有操作都设置在较小且固定的图像分辨率下，并在 SSNet 的末端使用超分辨率模块来恢复原始图像分辨率。因此，SVCNet 在推理时能够适应不同的图像分辨率。最后，与基于样例的视频着色相比，所提出的框架只需要输入稀疏颜色涂鸦。用户无需选择合适的样例图像，只需在某些特定像素上指定所需颜色。

我们在 DAVIS [20] 和 Videvo [21] 数据集上训练并评估 SVCNet。大量实验表明，SVCNet 的表现优于当前最先进的视频着色方法。我们还展示了 CPNet 在基于涂鸦的图像着色方面达到了最先进的性能，并减少了颜色渗漏伪影。本文的主要贡献如下：

1）我们提出了首个基于涂鸦的视频着色框架，称为 SVCNet，分别包含颜色传播和时空平滑两个阶段。我们将大多数操作设定在较小且固定的图像分辨率下，以降低生成不同高分辨率视频的计算成本；

2）我们提出了一种用于视频着色的时间聚合方法，涵盖短程和长程连接；

3）我们采用分割损失来解决视频着色领域中的颜色渗漏伪影。此外，当数据集中没有标注的分割图时，我们生成显著性图作为伪二值分割图。

II. 相关工作

图像着色。图像着色旨在从对应的灰度图像中重建颜色嵌入。图像着色可分为基于参考的着色（例如，基于涂鸦的着色 [22]–[33]，、基于示例的着色 [5]–[6]，[12]，[13]，[34]–[46]，、基于文本的着色 [47]–[49]）和全自动着色 [18]，[50]–[66]。基于参考的方法需要额外的用户输入，其中包含与期望颜色相关的信息。着色系统利用这些信息为灰度输入图像赋予可能的颜色。具体而言，基于涂鸦的方法会将用户给定的彩色涂鸦传播到图像的其余部分。基于示例的方法会将示例图像中的颜色赋予灰度图像。基于文本的方法则将文字或语言中的信息转换为颜色。近年来，深度神经网络凭借卓越的特征表示能力提升了着色效果。在此基础上，全自动方法可直接在大型数据集上学习端到端着色，无需任何额外信息。为进一步提升着色质量，研究者采用了一些特定设计，例如超列 [51]，、预训练骨干网络[51]–[53]，[57]，[59]，、多任务学习 [53]，[56]–[59]，以及辅助损失函数 [54]。

视频着色。如图 2 所示，根据数据流，现有的视频着色方法主要分为四类，具体描述如下：

1) 图像着色与时间平滑 [1]，[2]，[4]，[67]，[68]（图 2 (a)）：基于图像着色方法的进展，他们通过后处理为单个着色帧添加了时间一致性；

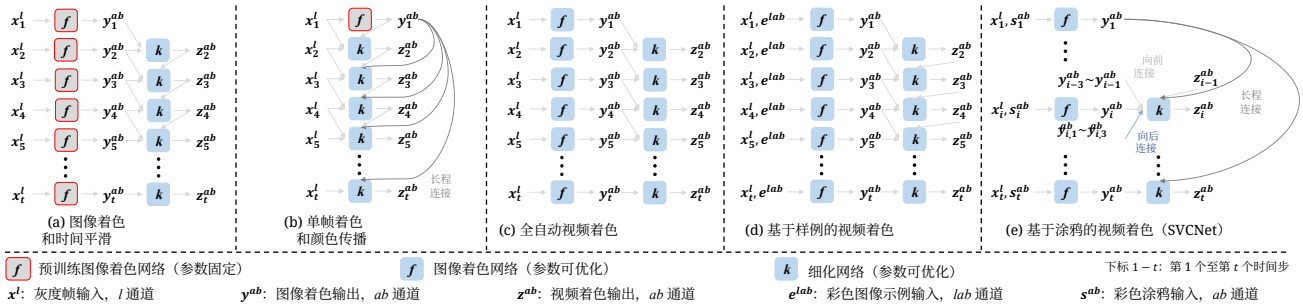


图 2. 视频着色方法数据流示意图：(a) 图像着色与时间平滑 [1]–[4]；(b) 单帧着色与颜色传播 [5]–[8]；(c) 全自动视频着色 [9]–[11]；(d) 基于示例的着色 [12]–[14]；(e) 基于涂鸦的着色（SVCNet），其中用户仅提供第一幅彩色涂鸦 s_1^{ab}。后续的彩色涂鸦 s_2^{ab}、 s_3^{ab}、...、 s_t^{ab}通过使用前向光流 $O_{1\to 2}$、 $O_{2\to 3}$、...、 $O_{t-1\to t}$对第一幅彩色涂鸦 s_1^{ab}进行变形而得到。

2) 图像着色（例如第一帧）与颜色传播 [5]–[8]（图 2 (b)）：他们使用一个着色帧来启动着色，后续帧的颜色及其对应关系则按顺序学习；

3) 全自动视频着色 [9]–[11]（图 2 (c)）：与类别 1) 和 2) 分别单独学习不同，这类方法利用神经网络在大型视频数据集上联合学习视频着色和时间对应关系；

4) 基于样例的视频着色 [12]–[14]（图 2 (d)）：该类方法将示例图像中的颜色传播到视频中的单色帧。

通常，类别 1) 和 2) 的效果都不理想，因为图像着色优化与时间平滑是分开进行的。这些方法需要强大的预训练图像着色算法，但着色后的视频通常在时间上不连贯。为了解决这一问题，另外两类方法将图像着色与时间平滑结合起来，并进行联合学习。例如，Lei 等人 [9] 提出了一种两阶段多模态视频着色框架。Kouzouglidis 等人 [10] 采用三维卷积作为基本算子。Zhao 等人 [11] 在生成器中使用了全局特征提取器和占位符特征提取器。尽管这些方法能够自动生成彩色视频，但其结果仍不够鲜艳。为提升着色质量，类别 4) 引入了额外的示例引导图像。例如，Zhang 等人 [12] 利用非局部网络将示例图像与灰度帧对齐，然后通过 ColorNet 将二者融合。这类方法需要高质量的示例图像才能获得满意的结果。为进一步简化样例选择，我们提出了首个基于涂鸦的视频着色方法，称为 SVCNet。SVCNet 的数据流如图 2 (e) 所示。

基于涂鸦的着色。基于涂鸦的着色旨在将用户提供的颜色涂鸦传播到单色图像。常见的传播方案会利用局部对应关系 [22], 边缘 [23], 或亮度加权色度融合 [24]。然而，这些方法关注局部关系，无法为远离颜色涂鸦的像素着色。为了建模长程连接，Xu等人 [27]提出了一种基于亲和度的图像编辑方案，Chen等人 [26] 则学习了特征空间中的映射。然而，结果仍然与给定颜色涂鸦的数量或位置密切相关

此外，当给定的颜色涂鸦靠近物体边缘时，颜色溢出现象十分明显。最近，Zhang等人 [30] 使用神经网络提取语义信息，取得了更好的效果。在本文中，我们进一步通过 SVCNet 框架将基于涂鸦的着色扩展到视频。

显著性检测。其旨在通过“显著图”定位图像中潜在的、具有感知显著性的区域。早期的显著性检测方法基于颜色变化 [69],、边界 [70],和超像素 [71], 等手工设计特征，能够预测出可信的边界，但无法准确刻画显著目标的结构。近期基于深度学习的方法将显著性检测推广至各种图像，并采用循环网络 [72], 、编码器-解码器 [73]–[76],和特征金字塔网络 [77]–[80] 等不同架构，融合细节与高层语义信息。

语义分割。其旨在定位图像中的不同目标。开创性的深度学习方法全卷积网络[81]采用了反卷积以及池化层融合。为增强特征表示和上下文信息，PSP-Net [82] 采用了金字塔池化模块，而 DeepLab[83] 则使用了不同膨胀率的空洞卷积。在此基础上，Chen 等人。 [84] 将空洞空间金字塔池化模块与 DeepLab 中的编码器-解码器架构相结合，使网络能够捕获跨层和层内特征。Yang 等人。 [85] 提出了密集空洞空间金字塔池化模块，以整合不同的空洞分支。近年来，三维分割 [86]–[88] 等其他数据模态中的视觉识别也取得了不错的性能。

III. 方法

A. 问题定义

给定连续的灰度帧，我们旨在基于用户提供的彩色涂鸦生成逼真的颜色嵌入。为辅助着色，我们还会生成相应的分割图，以尽量减少颜色渗漏伪影。我们将该问题表述为：在给定灰度帧 x_1^l， x_2^l.....x_t^l、用户提供的彩色涂鸦 s_1^{ab} 以及 SVCNet 的网络参数 Θ 的条件下，最大化颜色嵌入和分割图的后验概率：

$$\Theta^* = \arg\max_{\Theta} p(x_{1:t}^{ab}, q_{1:t} | x_{1:t}^l, s_1^{ab}, \Theta), \tag{1}$$

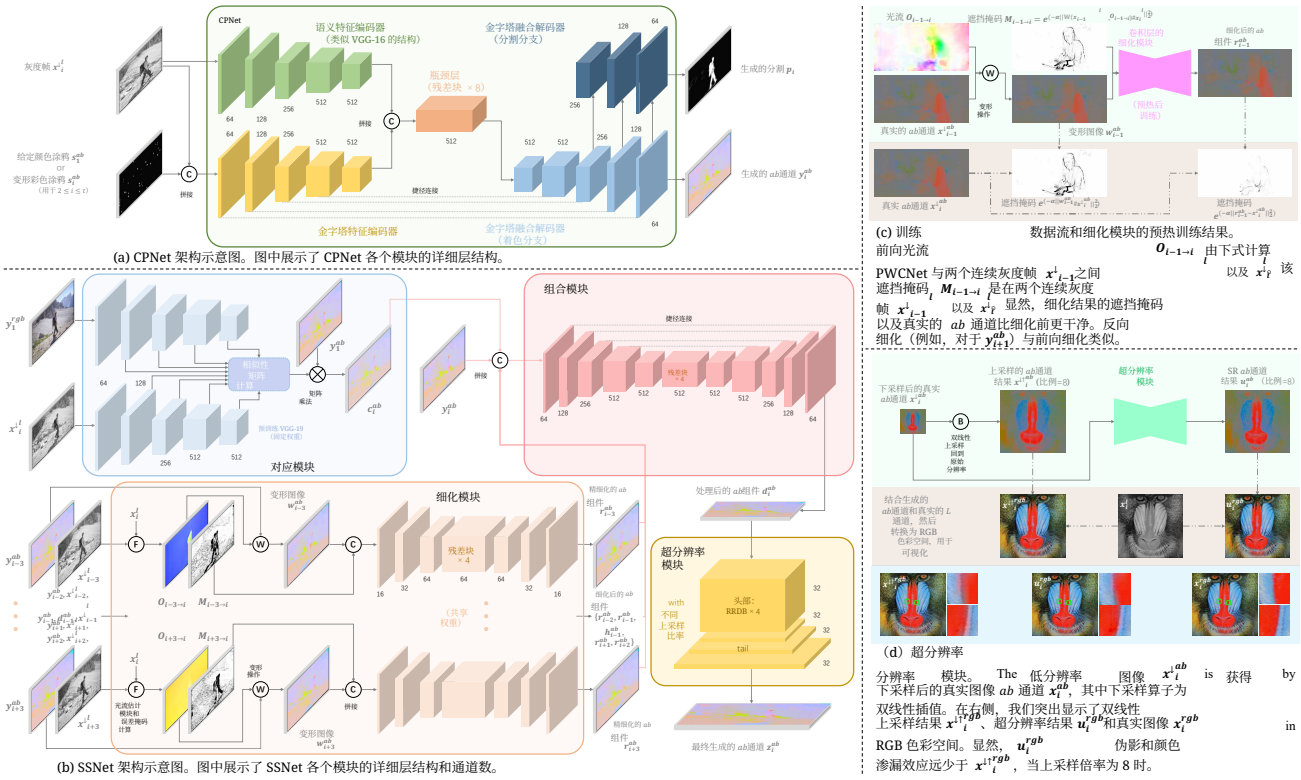


图 3. (a) CPNet 和 (b) SSNet 的详细架构示意图。图中标注了特殊块，其余块为普通卷积层。块旁标注了通道数。(c) SSNet 细化模块和 (d) SSNet 超分辨率模块的训练数据流及预热训练结果示意图。

其中， $x^l_{1:t}$ 和 $x^{ab}_{1:t}$ 分别是从第 1 帧到第 t 帧的 *CIELab* 彩色空间中的灰度分量和颜色分量。 $q_{1:t}$ 是真实分割图。Θ^* 是理论上最优的 SVCNet 参数。为优化网络，我们进一步针对颜色嵌入和分割图定义损失函数 L：

$$\underset{SVCNet(.;\Theta)}{minimize} \sum_{i=1}^{t}(L(z_i^{ab}, x_i^{ab}) + L(p_i, q_i)), \tag{2}$$

其中，z_i^{ab} 和 p_i 是 SVCNet 在时间步 i 生成的输出。x_i^{ab} 和 q_i 分别是时间步 i 的真实值颜色嵌入和分割图。$SVCNet$（。； Θ）表示网络参数为 Θ 的 SVCNet。

*B. SVCNet*架构

SVCNet由两个子网络组成：彩色传播子网络（CPNet）和时空平滑子网络（SSNet），如图1和图3所示。下面介绍各个子网络和模块。

CPNet。CPNet 执行基于涂鸦的图像着色，其包含两个编码器和两个解码器，如图 3 (a) 所示。为融合灰度输入和彩色涂鸦引导的信息，我们采用金字塔特征编码器。考虑到通过变形首个彩色涂鸦得到的远时间步彩色涂鸦可能会消失，我们采用另一个预训练网络——语义特征编码器来提升着色质量。所得特征经拼接后由瓶颈层进行融合。随后，金字塔特征解码器根据瓶颈层的输出和金字塔特征编码器的捷径连接生成颜色嵌入 [89]。此外，最后三个解码器层的输出会输入分割分支，以生成相应的分割图。在训练过程中，分割分支通过反向传播帮助修正着色分支的权重。

SSNet。SSNet对CPNet的输出进行后处理。它增强生成帧的时间一致性，并包含四个模块，如图 3（b）所示。

细化模块。该模块旨在处理短程相关性，这种相关性由时间步 i 上的前序帧（y^{ab}_{i-3}， y^{ab}_{i-2}， y^{ab}_{i-1}， d^{ab}_{i-1}）和后续帧（y^{ab}_{i+1}， y^{ab}_{i+2}， y^{ab}_{i+3}）共同建模。为了使它们的位置与当前帧匹配，我们首先利用光流将它们变形至当前帧的位置，光流由预训练的 PWC-Net [90] 在灰度帧上计算得到。随后，我们使用细化模块对它们进行后处理，以尽量减少由遮挡和运动边界造成的变形伪影。为了表示这些区域，我们计算变形后的前序帧和后续帧的遮挡掩码，其定义如下：

$$M_{j\to i} = exp(-\alpha||\mathbb{W}(x^{\downarrow l}_j, O_{j\to i}) - x^{\downarrow l}_i||_2^2), \tag{3}$$

其中， $M_{j\to i}$ 和 $O_{j\to i}$ 分别表示从时间步 j 到 i的遮挡掩码和光流。 $x^{\downarrow l}_j$ 和 $x^{\downarrow l}_i$ 分别是时间步 j 和

i时下采样的真实灰度图像。$\mathbb{W}(*)$表示变形操作。α 是一个控制遮挡掩码敏感度的超参数。细化模块（RM）的前向过程可表示为：

$$r_j^{ab} = \mathrm{RM}(\mathbb{W}(y_j^{ab}, O_{j\to i}), M_{j\to i}), \tag{4}$$

其中，我们设置 $j = (i-3,\ i-2,\ i-1,\ i+1,\ i+2,\ i+3)$。此外，我们按如下方式细化前一个 SSNet 的低分辨率输出 d_{i-1}^{ab}，该输出是组合模块的输出：

$$h_{i-1}^{ab} = \mathrm{RM}(\mathbb{W}(d_{i-1}^{ab}, O_{i-1\to i}), M_{i-1\to i}). \tag{5}$$

对应模块。对应模块对第一张着色帧与当前帧之间的长程连接进行建模。由于这两帧之间通常存在较大的运动，我们不使用光流。相反，我们计算它们之间的相似度矩阵，并用其扭曲第一张生成帧的颜色嵌入特征。首先，与 [12]，中的做法类似，我们通过提取预训练 VGG-19 网络 [91] 的第 $conv_{2_2}$、$conv_{3_2}$、$conv_{4_2}$和 $conv_{5_2}$层特征，构建 RGB格式的首帧 y_1^{rgb} 和当前灰度帧 $x_i^{\downarrow l}(y_1^{rgb}, x_i^{\downarrow l} \in \mathbb{R}^{H\times W\times C})$ 的特征金字塔。所有特征均上采样至与 $conv_{2_2}$ 相同的分辨率（即 $\frac{1}{2}H \times \frac{1}{2}W$），然后拼接在一起。$y_1^{rgb}$ 和 $x^{\downarrow l}_i$ 的结果特征分别记为 F_1 和 F_i。其次，相似度矩阵 $S_{1\leftrightarrow i} \in \mathbb{R}^{\frac{1}{4}HW\times\frac{1}{4}HW}$ 的计算方式如下：

$$S_{1\leftrightarrow i} = \frac{F_1 - \mu(F_1)}{||F_1 - \mu(F_1)||_2} \cdot \frac{F_i - \mu(F_i)}{||F_i - \mu(F_i)||_2}, \tag{6}$$

其中，$\mu(*)$表示均值操作。然后，我们使用相似度矩阵按如下方式对 y_1^{ab} [12] 进行变形：

$$c_i^{ab} = (\sum_m softmax(\tau \cdot S_{1\leftrightarrow t}(:, m)) \cdot y^{\downarrow ab}_1)^{\uparrow}, \tag{7}$$

其中，c_i^{ab}是第一个生成的颜色嵌入y_1^{ab}经过变形后的结果。τ 是温度参数。由于需要匹配相似度图与颜色嵌入的分辨率，因此进行了下采样操作。

组合模块。接下来，组合模块通过类U-Net架构 [89]，聚合来自细化模块输出（$r_{i-3}^{ab}, r_{i-2}^{ab}, r_{i-1}^{ab}, h_{i-1}^{ab}, r_{i+1}^{ab}, r_{i+2}^{ab}, r_{i+3}^{ab}$）、对应模块（$c_i^{ab}$）以及 CPNet 当前生成的颜色嵌入（$y_i^{ab}$）中的信息。组合模块的输出记为 d_i^{ab}。

超分辨率模块。最后，由于先前的操作均在固定的小分辨率下运行，超分辨率模块会在推理时根据d_i^{ab}将帧恢复至原始尺寸。对于不同的高分辨率，我们使用相同的特征提取头，但采用具有多个上采样倍率的不同尾部。最终输出记为 z_i^{ab}。

C. 训练策略

直接优化规模庞大的 SVCNet 而不进行任何初始化，很容易遇到梯度爆炸问题。为稳定训练，我们分别对 CPNet 和 SSNet 的部分模块进行预热预训练。之后，我们在视频数据集上训练完整的 SVCNet。具体如下：

1) CPNet 预热预训练：我们在 ImageNet数据集 [15]，上对 CPNet 进行预训练，该数据集比视频数据集包含更多的模式和场景。由于 ImageNet数据集不提供分割图，我们通过 [78]生成显著性图作为真实值。随后，我们使用视频数据集（DAVIS [20]和 Videvo [21]）中的单帧对 CPNet 进行微调，使其更好地适应视频帧的尺寸。同样，我们通过 [78]为 Videvo 数据集生成显著性图，作为伪分割图。

2) SSNet 预热预训练：我们在两个视频数据集 DAVIS [20]和 Videvo [21]上，对细化模块和超分辨率模块进行自监督学习。首先，对于视频中的一帧 x_i^{ab}，我们随机将前一帧 x_{i-1}^{ab} 或后续帧 x_{i+1}^{ab} 扭曲至 x_t^{ab}的位置，作为细化模块的输入。我们让该模块重建自身，并使用 $L1$ 损失进行训练。其次，我们以不同倍率（2、4 和 8）对原始分辨率的一帧进行下采样，作为超分辨率模块的输入。然后，我们让超分辨率模块重建自身，并使用 $L1$ 损失进行训练。预热训练仅针对*ab* 彩色分量进行。当损失足够小且稳定时，训练便结束。结果如图 3 (c) 和 (d) 所示。细化模块可以减少遮挡区域中的伪影。超分辨率模块可以最大程度地减少颜色渗漏伪影，尤其是在上采样倍率较大时（例如 8）。

3）联合训练阶段：我们基于所有预热训练权重优化完整的 SVCNet（CPNet 和 SSNet）。我们从视频中随机选取连续的 7 帧，并在训练时将前 6 帧翻转后作为一个批次。接下来，我们将介绍“CPNet 预热预训练”和“联合训练阶段”的损失函数。

D. 损失函数

训练 CPNet 时，我们分别对两个金字塔融合解码器分支的输出采用着色损失和分割损失：

$$L_c^{CP} = \mathbb{E}[||y_i^{ab} - x^{\downarrow ab}_i||_1], \tag{8}$$

$$L_s^{CP} = \mathbb{E}[||p_i - q_i||_1], \tag{9}$$

其中，y_i^{ab}和 p_i 是两个分支的输出。$x^{\downarrow ab}_i$ 和q_i 是对应的真实值。然后，CPNet 预热训练的总损失函数定义为：

$$L^{CP} = L_c^{CP} + \lambda_s L_s^{CP}, \tag{10}$$

其中，λ_s 是权衡参数。

对于联合训练，我们保留 CPNet 损失函数，并加入 SSNet 损失函数。具体而言，我们分别对组合模块（d_i^{ab}）的输出和超分辨率模块（z_i^{ab}）的输出采用着色损失。它们可表示为：

$$L^{SS} = \mathbb{E}[||d_i^{ab} - x^{\downarrow ab}_i||_1] + \mathbb{E}[||z_i^{ab} - x_i^{ab}||_1], \tag{11}$$

其中，$x^{\downarrow ab}_i$ 和 x_i^{ab}分别是低分辨率和原始分辨率下的真实值。

联合训练阶段的完整损失函数为以下各项之和：

$$L^{joint} = L^{CP} + L^{SS}. \tag{12}$$

表 I 不同训练阶段的优化细节总结。

训练阶段	训练网络	Loss	训练集	总迭代次数	初始学习率 (LR)	学习率减半迭代次数
1) CPNet 预热	CPNet	L^{CP}	ImageNet	800,740	1×10^{-4}	400,370
预训练	CPNet	L^{CP}	DAVIS+Videvo	200,600	5×10^{-5}	100,300
2）SSNet 预热	细化模块	$L1$	DAVIS+Videvo	312,000	1×10^{-4}	156,000
预训练	超分辨率模块	$L1$	DAVIS+Videvo	296,200	1×10^{-4}	148,100
3) 联合训练阶段	CPNet+SSNet	L^{joint}	DAVIS+Videvo	312,000	CPNet:1×10^{-6}; SSNet:5×10^{-5}	156,000

IV. 实验

A. 实现细节

训练数据集。 我们使用完整的 ImageNet数据集进行 CPNet 预热预训练。该数据集包含 130 万张图像，分为 1000 个类别。其他训练阶段使用 DAVIS 和 Videvo 数据集，其中包含 156 个短视频片段，共 29620 帧。DAVIS数据集提供带标注的二值分割图。对于 ImageNet 和 Videvo 数据集，我们生成显著性图作为伪分割图。预处理时，将 ImageNet 图像调整至 256×256 分辨率。将 DAVIS 和 Videvo 数据集中的视频帧调整至 256×448，作为 SVCNet 的输入；并调整至 512×896 ，作为真实值（即在联合训练阶段将超分辨率模块的上采样倍率设为 2）。所有图像均归一化至 [0, 1] 范围。

彩色涂鸦。 我们从真实值图像的*ab*通道中随机选取彩色涂鸦。训练时有一半概率使用有效的彩色涂鸦，且彩色涂鸦的数量为 1 至 40 个。在联合训练阶段，一个批次中仅为第一帧提供彩色涂鸦。后续帧的彩色涂鸦通过前向光流对前一帧的涂鸦进行变形得到。

优化。 优化细节总结于表 I 中，其中分别列出了每个训练阶段的训练网络、损失函数、训练集、总训练迭代次数、初始学习率（LR），以及学习率减半时对应的具体迭代次数。在联合训练阶段开始时，我们使用预热预训练权重初始化 CPNet 和 SSNet。对于其余层或块，我们使用 [92] 对其进行初始化。预热预训练阶段的批量大小为 4，联合训练阶段每个图形处理器的批量大小为 1。我们使用 Adam 优化器 [93] ，其中 β_1=0.5、 β_2=0.999。权衡参数 λ_s 设为 0.1。掩码参数 α 和温度参数 τ 均为 200。我们使用 PyTorch 框架实现 SVCNet，并在 8 块 NVIDIA V100 图形处理器和 8 块 NVIDIA Titan Xp 图形处理器上进行训练。考虑到并行训练，完成预热预训练约需 10 天，完成联合训练还需 6 天。

网络架构。SVCNet架构如图 3 所示，我们着重标示了捷径连接 [89], 残差块(ResBlock) [94], 以及残差中的残差密集块 (RRDB) [95]等特殊模块。我们还标示了通道数。除第一层和第二层外，我们均使用 LeakyReLU [96]作为激活函数。我们仅在细化模块中使用实例归一化 [97]。

B. 定量指标

生成质量。 我们采用峰值信噪比和结构相似性 [98] 分别计算生成结果与真实值之间的像素级准确率和结构相似性。对于基于涂鸦的着色任务，验证阶段会提供一些真实涂鸦，因此着色任务变成了特定任务。因此，峰值信噪比和结构相似性是评估生成质量的合适指标。

分割性能。我们使用 HRNetV2 + OCR [99],[100] 计算 DAVIS 语义分割验证数据集 [20]上生成帧的平均交并比（mIoU）。如果某种方法获得较高的 mIoU 值，那么它遇到颜色溢出问题的可能性可能较低，因为分割算法能够更好地从该方法生成的着色图像中分离关键对象。**人类偏好**。我们对不同方法的视频着色结果开展人类偏好研究。从 DAVIS 和 Videvo 验证集中随机选取了 10 个视频，供用户进行比较。对于每个视频，人类观察者需要根据时间一致性和色彩鲜艳度选出最佳结果。实验共有 10 名人类观察者，他们可以多次观看这些视频。

C. 视频着色实验

实验设置。我们将 SVCNet 与其他目标相似的近期方法的视频着色性能进行比较，这些方法可分为四种流水线:

1) 图像着色和时间平滑（如图 2 (a) 所示）： CIC [52], LTBC [53],ChromaGAN [57], 和 IAC [58] 用作图像着色算法；BTC [2]和 DVP [4] 用作时间平滑网络；

2) 图像着色和颜色传播（如图 2 (b) 所示）： DEVC [12] 用于将第一个着色帧传播到后续帧；

3) 全自动视频着色（如图 2 (c) 所示）： FAVC [9], 3 DVC [10], 和 VCGAN [11];

4) 基于涂鸦的图像着色与时间平滑/颜色传播: RUIC [30]用作基于涂鸦的图像着色方法；DVP [4] 和 DEVC[12] 用作后处理方法。请注意，RUIC 在视频数据集上进行了微调。

我们在 DAVIS [20] 和 Videvo [21]验证集上进行实验。总共有 50 个视频片段，每个片段约包含 100 帧。在实验中，我们为基于涂鸦的方法使用 40 个颜色涂鸦，这些涂鸦从第一帧中随机提取。后续涂鸦通过对第一

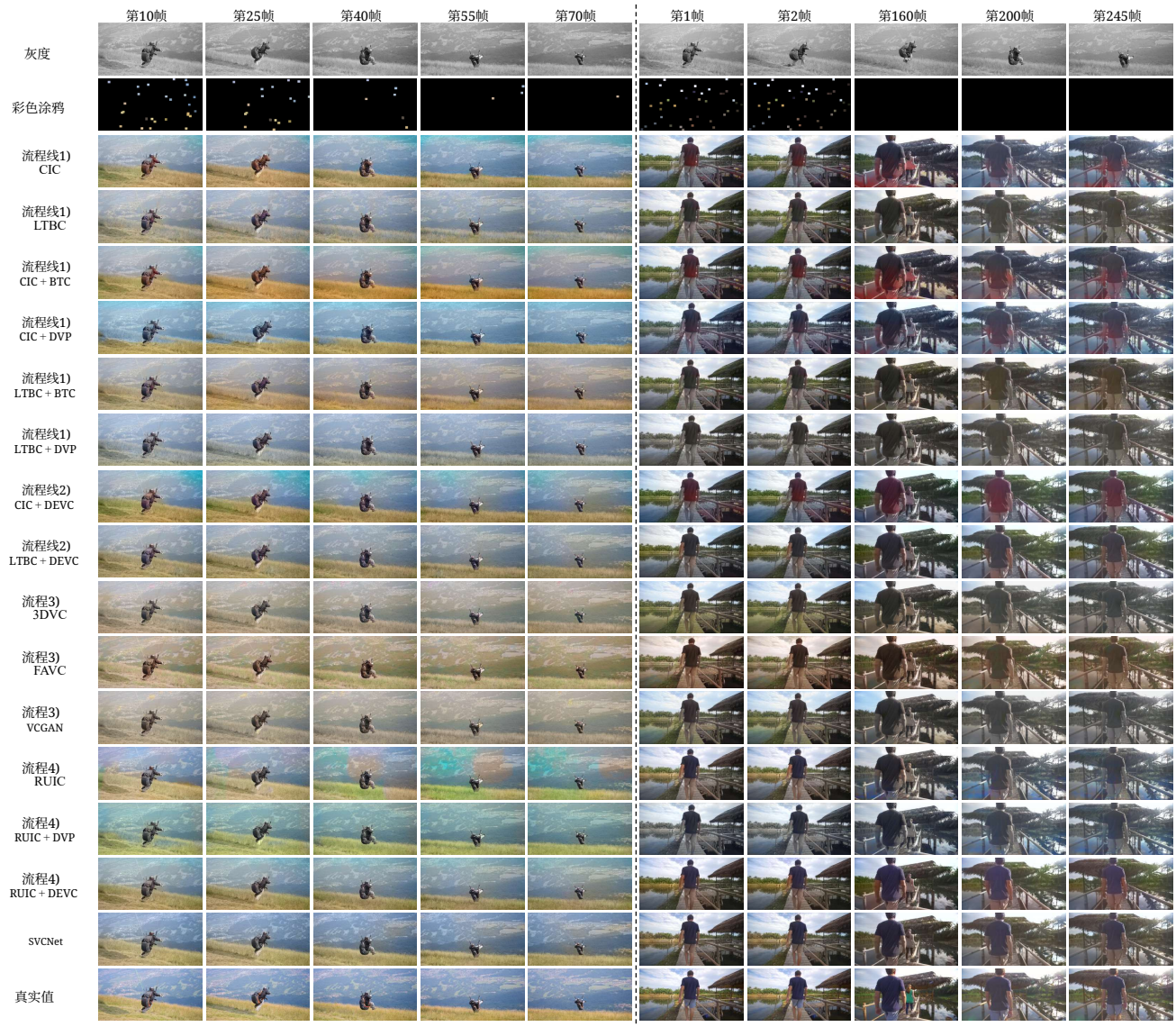


图 4. SVCNet 与其他流水线的视频着色结果示例。图像分别提取自 DAVIS数据集中的“paragliding-launch”样本和 Videvo 数据集中的“YogaHut2”样本。

帧的颜色涂鸦应用前向光流进行变形得到。不同的流水线共享相同的颜色涂鸦。对于 SVCNet，我们将超分辨率模块的上采样比例设为 2，以获得分辨率为 512×896的结果，然后将其调整为 DAVIS 和 Videvo 数据集的目标分辨率。其他方法则以与验证图像相同的分辨率运行。DVP 的整体调优周期默认设为 25。请注意，FAVC 和 DVP 会自动裁剪输入图像，使其边长成为 32 的倍数。因此，我们仅使用有效区域计算定量指标。定量结果汇总于表 II，部分示例见图4和图5。

着色质量。 首先，从图 4 可以看出，在色调和色彩自然度方面，SVCNet 的结果比其他方法的结果更接近真实值。例如，在“paragliding-launch”样本中，3DVC、FAVC 和 VCGAN 的结果色彩较淡。这些方法未能很好地平衡视频着色的鲜艳度与时间平滑度，也就是说，它们的结果过于平滑，色彩不够鲜艳。尽管 BTC、DVP 和 DEVC 能够平滑单帧着色图像，但结果仍不够合理。例如，在“YogaHut2”样本中，与 CIC 相关的结果存在明显的色彩伪影，例如衣服被着成红色。由于 CIC 会为图像中的每个像素预测一个颜色分布，因此与真实值相比，其输出并不总是自然。

其次，与其他方法相比，SVCNet 的结果中颜色溢出问题较少。例如，在“paragliding-launch”样本中，CIC + DVP 将人类着色为蓝色，因为它无法区分人类和天空。在“YogaHut2”样本中，3DVC 将手着色为绿色，因为周围树木的绿色影响了对手的着色。与其他方法相比，SVCNet 的颜色渗漏伪影少得多，因为它使用了分割

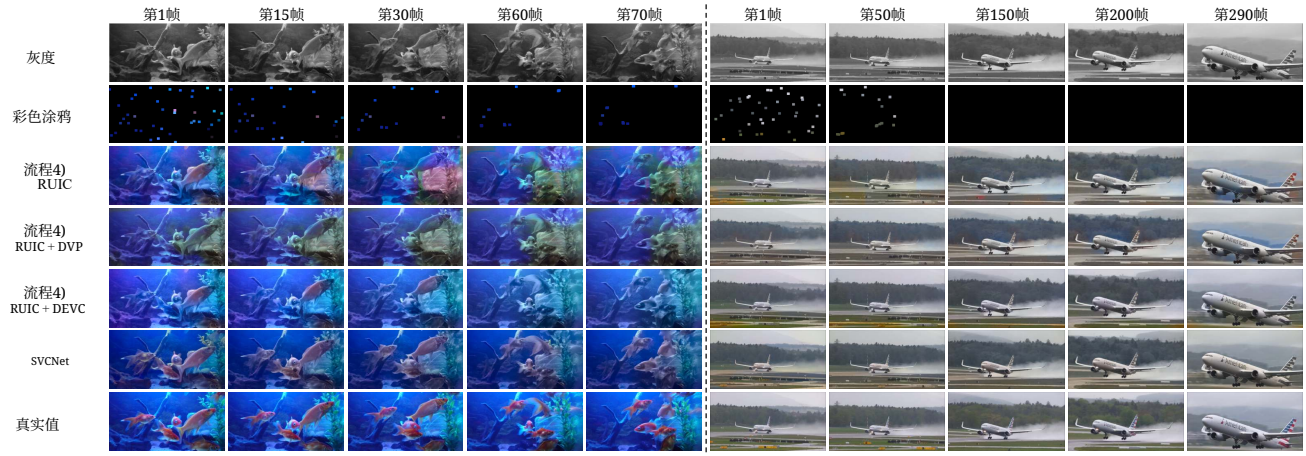


图 5. 视频着色结果示意图 r DAVIS数据集的“金鱼”样本 展示了 SVCNet 与最先进的基于涂鸦的视频着色方法的结果。这些图像提取自以及 Videvo 数据集的“AircraftTakingOff1”样本。

表 II C对比视频着色流水线与所提出的 SVCNET 在 DAVIS 和 VIDEVO 数据集。 THE 红色、蓝色、和绿色分别表示最佳、次佳和第三佳表现。“IC”和“VC”分别表示“图像着色方法”和“视频着色方法”。

方法	Type	彩色涂鸦	DAVIS 峰值信噪比 ↑	DAVIS 结构相似性 ↑	DAVIS 平均交并比 ↑	Videvo 峰值信噪比 ↑	Videvo 结构相似性 ↑
1) CIC [52]	IC	/	22.48	0.9043	0.8419	21.83	0.9024
1) LTBC [53]	IC	/	23.96	0.9171	0.8945	24.69	0.9272
1) ChromaGAN [57]	IC	/	23.77	0.9420	0.8879	23.94	0.9392
1) IAC [58]	IC	/	22.88	0.9422	0.8836	23.99	0.9486
1) 图像压缩 + BTC [2], [52]	VC	/	21.50	0.8935	0.8419	21.05	0.8833
1) LTBC + BTC [2], [53]	VC	/	22.47	0.9044	0.8899	22.83	0.9105
1) ChromaGAN + BTC [2], [57]	VC	/	19.91	0.8938	0.8796	16.64	0.8325
1) IAC + BTC [2], [58]	VC	/	18.40	0.8778	0.8698	15.85	0.8209
1) CIC + DVP [4], [52]	VC	/	23.30	0.9351	0.8724	22.23	0.9328
1) LTBC + DVP [4], [53]	VC	/	24.06	0.9425	0.8886	24.75	0.9548
1）ChromaGAN + DVP [4], [57]	VC	/	23.81	0.9444	0.8800	23.97	0.9451
1）IAC + DVP [4], [58]	VC	/	22.91	0.9407	0.8936	23.99	0.9503
2) CIC + DEVC [12], [52]	VC	/	21.64	0.9321	0.8661	21.36	0.9231
2) LTBC + DEVC [12], [53]	VC	/	22.46	0.9397	0.8688	24.03	0.9513
2) ChromaGAN + DEVC [12], [57]	VC	/	22.56	0.9429	0.9026	22.40	0.9386
2) IAC + DEVC [12], [58]	VC	/	22.43	0.9384	0.8712	23.59	0.9479
3) 3DVC [10]	VC	/	23.49	0.9151	0.8948	24.33	0.9231
3) FAVC [9]	VC	/	22.98	0.9055	0.8889	23.47	0.9183
3) VCGAN [11]	VC	/	23.43	0.9133	0.8954	24.73	0.9225
4) RUIC [30]	IC	✓	25.42	0.9456	0.8995	25.02	0.9432
4) RUIC + BTC [2], [30]	VC	✓	21.16	0.8869	0.8900	17.07	0.8279
4) RUIC + DVP [4], [30]	VC	✓	25.82	0.9455	0.9075	24.68	0.9460
4) RUIC + DEVC [12], [30]	VC	✓	24.85	0.9524	0.9002	25.66	0.9583
SVCNet	VC	✓	25.71	0.9565	0.9104	26.30	0.9615

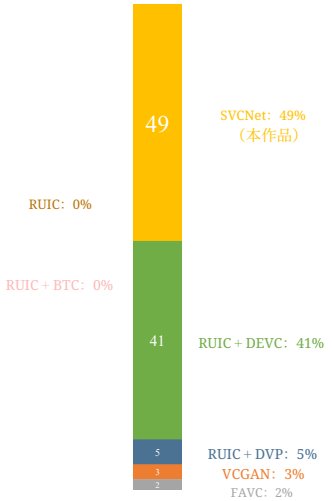


图6. 人类偏好研究结果。图中标出了各方法的人类偏好率。不同颜色代表不同方法。

分支来缓解这一问题。

第三，SVCNet 比其他方法具有更好的视频时间一致性，因为它聚合了短程连接和长程连接。它的时间一致性优于其他方法。对于流程线1)，图像着色和时间平滑方法生成的视频不够连贯，例如 CIC+ BTC、CIC + DVP、LTBC + BTC 和 LTBC + DVP。由于图像着色和时间平滑是分别学习的，最终输出仍然接近单帧着色结果。例如，在 CIC 结果中，第160帧的人物被着色为红色，而到了第200帧则变成灰色；但 CIC +BTC 无法缓解这一问题，即结果的时间一致性仍然不够好。对于流程线2)，视频比流程线1)的结果更平滑。然而，由于图像着色方法和 DEVC 并未联合训练，因此仍存在类似问题。对于流程线3)，与其他方法相比，视频连续性较好，但流程线3)的结果色彩较少。对于流程线4)，RUIC + DEVC 优于另外两种方法 RUIC 和 RUIC +DVP。然而，它仍存在基于 DEVC 的方法的常见问题，即结果高度依赖于第一帧着色帧。

最后，与 RUIC 相比，SVCNet 能更好地利用给定的颜色涂鸦。由于只提供了第一帧的颜色涂鸦，它们可能会在较远的帧中消失。在这种情况下，RUIC

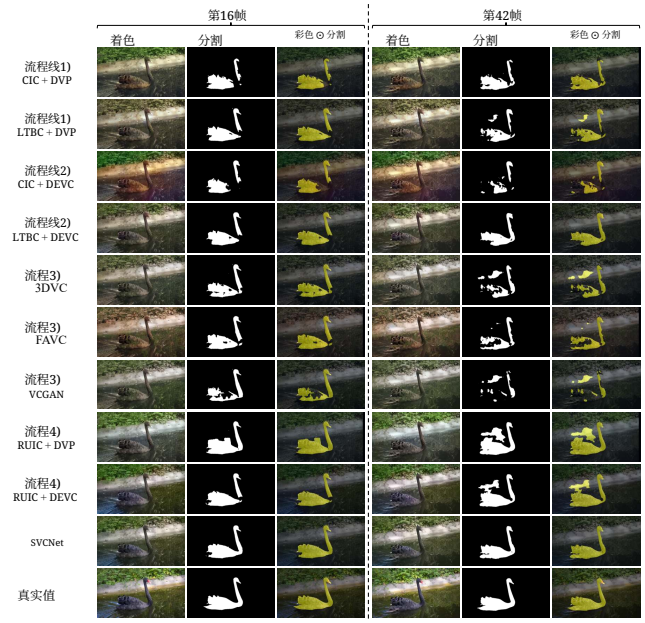


图 7. 展示了 SVCNet 及其他流水线在 DAVIS数据集的“blackswan”样本上的视频着色和分割结果。



图 8. SVCNet 在不同颜色涂鸦下的结果示意图。图像分别取自 DAVIS数据集中的“blackswan”样本和 Videvo 数据集中的“CoupleRidingMotorbike”样本。省略了输入彩色涂鸦。

无法生成合适的结果，例如，在“paragliding-launch”样例中，RUIC 的结果存在明显的闪烁伪影。尽管 DVP 和 DEVC 能为 RUIC 的结果带来时间一致性，但这些结果与真实值仍相差甚远。我们在图 5 中展示了更多示例。与真实值相比，RUIC + DVP 和 RUIC + DEVC 的结果在色彩鲜艳度和准确度方面均不如 SVCNet。

着色保真度。 根据表 II，SVCNet 在峰值信噪比和结构相似性方面均优于其他方法。这表明 SVCNet 能够充分利用给定颜色涂鸦中的颜色。与最先进的方法 RUIC + DEVC 相比，SVCNet 在 SSNet 中额外采用了短程连接；与 RUIC + DVP 相比，SVCNet 则额外采用了长程连接。短程连接包括前面三帧和后面三帧的单帧着色结果，以及最后一个输出。长程连接是来自第一帧着色结果的信息。由于 SVCNet 聚合了短程信息、长程信息和当前时间步的输出，因此获得了更高的着色保真度。

人类偏好。 我们在人类偏好研究中选取了表现最佳的方法 FAVC、VCGAN、RUIC、RUIC + BTC、RUIC + DVP、RUIC+ DEVC 以及 SVCNet。结果如图 6 所示。显然，与其他方法相比，本文提出的 SVCNet 获得了更高的人类偏好率。实验表明，与其他结果相比，SVCNet 的上色结果在时间上更加连贯，色彩也更加丰富。

颜色渗漏分析。 本小节将进一步讨论颜色渗漏伪影。如图 7 所示，分割算法从 SVCNet 的结果中生成了最清晰的分割图。比如，与其他方法相比，SVCNet 中天鹅的轮廓更加连贯。这表明其关键对象与其他对象的区分度更高。此外，如表 II 所示，SVCNet 在所有方法中取得了最高的平均交并比。这说明其结果中的对象比其他方法更加明显，因此 SVCNet 出现颜色渗漏伪影的可能性更低。这是通过在训练时加入额外的分割损失实现的，着色分支的权重由分割分支进行调整。

着色多样性。由于颜色涂鸦具有多样性，基于涂鸦的视频着色应是一种多模态应用。我们在图 8 中展示了不同颜色涂鸦下的一些结果。显然，SVCNet 能够根据不同的输入彩色涂鸦生成多样的色彩。

D. 消融研究

实验设置。 为了展示 SVCNet 各组件的有效性，我们定义了七种消融研究设置。除各设置中对应的消融项目外，训练超参数均保持不变。在 DAVIS 和 Videvo 数据集上以原始分辨率进行评估，所有设置使用相同的颜色涂鸦。具体如下：

1) 不使用分割损失：去除分割损失 L_s^{CP}；2) 不进行 CPNet 预训练：去除在视频数据集上的 CPNet 预热阶段（即仅在 ImageNet 上进行预训练）；

3) 不使用短程连接：去除 SSNet 的细化模块中的所有短程连接（即 y_{t-3}^{ab}、y_{t-2}^{ab}、y_{t-1}^{ab}、y_{t+1}^{ab}、y_{t+2}^{ab} 和 y_{t+3}^{ab}，以及 z_{t-1}^{ab}）；

4）不使用长程连接：移除带有长程连接的对应模块（即，y_1^{ab}）；

表 III SVCNET 消融研究，针对DAVIS 和VIDEVO 数据集。最佳结果以红色表示。。

消融研究设置	对比项	彩色涂鸦	DAVIS 峰值信噪比 ↑	DAVIS 结构相似性 ↑	DAVIS 平均交并比 ↑	Videvo 峰值信噪比 ↑	Videvo 结构相似性 ↑
1) 不使用分割损失	训练方案	✓	24.09	0.9378	0.8780	25.04	0.9464
2）不使用 CPNet 预训练	训练方案	✓	19.19	0.8913	0.6464	20.27	0.9067
3）不使用短程连接	时间聚合	✓	24.60	0.9495	0.8949	25.04	0.9535
4）不使用长程连接	时间聚合	✓	24.23	0.9479	0.8919	24.51	0.9511
5) 无短程和长程连接	时间聚合	✓	24.01	0.9481	0.8934	24.37	0.9498
6) 使用 64×128分辨率	分辨率	✓	24.47	0.9522	0.8768	25.26	0.9576
7) 使用 128×224 分辨率	分辨率	✓	24.59	0.9533	0.8960	25.64	0.9589
SVCNet	完整的 SVCNet	✓	25.71	0.9565	0.9104	26.30	0.9615

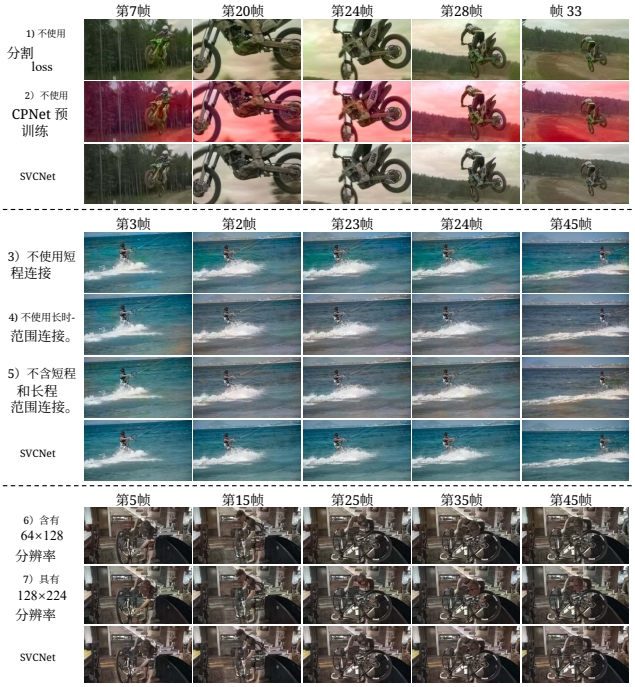


图 9. 不同消融研究设置的示意图。样本分别选自 DAVIS数据集中的“越野摩托车跳跃”“风筝冲浪”和“自行车旅行”视频。

5）不使用短程和长程连接：移除所有短程连接和长程连接；

6） 64×128 分辨率：将 SVCNet 的输入图像分辨率更改为 64×128。超分辨率模块的上采样倍率更改为 8；

7） 128×224 分辨率：将 SVCNet 的输入图像分辨率更改为 128×224。超分辨率模块的上采样倍率更改为 4。

定量结果汇总于表 III，部分样例如图 9 所示。

训练方案。 对于设置 1) 和 2)，我们在训练阶段省略了部分内容。若去掉分割损失，就会出现颜色渗漏伪影（例如，天空的某些区域被着成绿色），如图 9 所示。若在视频数据集上去掉 CPNet 预热预训练阶段，着色结果就会出错。在没有预热预训练的情况下训练所有模块极其困难，这是因为 CPNet 无法很好地为分辨率不同于图像的视频帧着色

表 IV SVCNET 在分辨率为 256×448 OR 1024×1792 ($MACs_{256}$ AND $MACs_{1024}$) 的图像块上的参数数量（N_{param}）和乘累加操作数（MACS）对比。

模块	N_{param}	$MACs_{256}$	$MACs_{1024}$
CPNet	91.475M	251.690G	251.690G
对应模块	26.942M	162.102G	162.102G
细化模块	346.608K	28.664G	28.664G
组合模块	7.084M	20.479G	20.479G
超分辨率模块	306.432K	33.096G	55.292G
SSNet	34.678M	244.340G	266.537G
SVCNet	126.153M	496.030G	518.226G

；与此同时，如果 CPNet 输出的着色帧效果不佳，SSNet 也会失效。此外，正如表 III 所示，设置 1) 的平均交并比以及设置 2) 的所有指标结果都明显下降。

时间聚合。 对于设置 3–5)，我们没有使用某些时间信息。如图 9 所示，设置 3) 得到的第 22–24 帧颜色不一致，因为没有使用短程连接。设置 4) 和 5) 得到的远处帧的颜色也与第 3 帧的结果不一致。由于没有使用长程连接，且在存在大幅运动时颜色涂鸦很容易消失，因此远处帧的颜色会失真。此外，表 III 显示，排除时间信息会导致峰值信噪比和结构相似性均大幅下降。实验结果表明，短程连接和长程连接对于 SVCNet 都至关重要。

分辨率。 对于设置 6) 和 7)，我们调整了图像分辨率。虽然降低分辨率可以提升推理速度，但 SVCNet 的性能会明显下降。此外，在大多数情况下，着色应用并不需要非常快的推理速度。综合考虑着色质量、推理速度和内存成本，我们将 256×448 用作运行分辨率。

总之，所有提出的训练方案、时间聚合和网络架构都具有重要意义。

E. 计算成本

SVCNet 的计算成本汇总于表 IV。基于颜色分量的稀疏性，大多数操作都在固定的小分辨率下进行（即，256×448）。因此，除超分辨率模块外，所有模块在不同输入分辨率下的计算成本均保持不变。当输入分辨率从

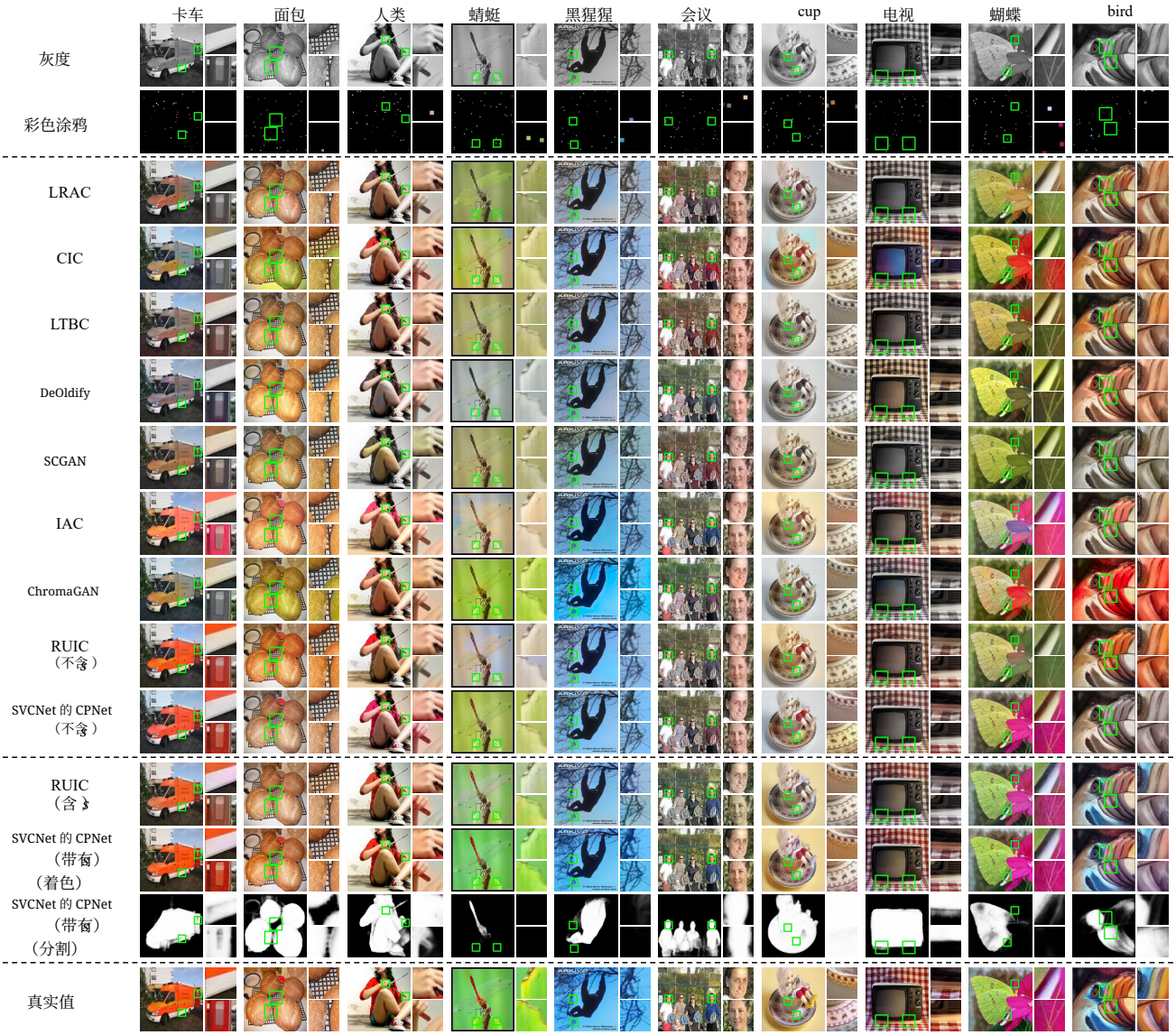


图 10. SVCNet 与最先进方法的图像着色结果示意图，其中 RUIC [30] 和 SVCNet 是基于涂鸦的方法。第一行和第二行分别表示灰度涂鸦和彩色涂鸦。最后一行和倒数第二行分别为真实值以及 SVCNet 预测的分割图。其他行展示了不同方法的着色结果。图像块与全分辨率图像并列显示。

从 1024×1792变为 256×448时，乘加运算次数（乘累加操作数）仅增加 22.197G，这仅为256×448 分辨率下乘加运算次数（496.030G）的 4.5%。这种设计使 SVCNet 内存友好。此外，细化模块和超分辨率模块采用小型架构便已足够。两者的参数总和不到整个 SVCNet 总参数的 1%。请参阅图 3 (b)，它们的基础通道数分别为 16 和 32。尽管采用了小型架构，但如图 3 (c) 和 (d) 所示，它们仍能很好地实现目标。

F. 图像上色实验

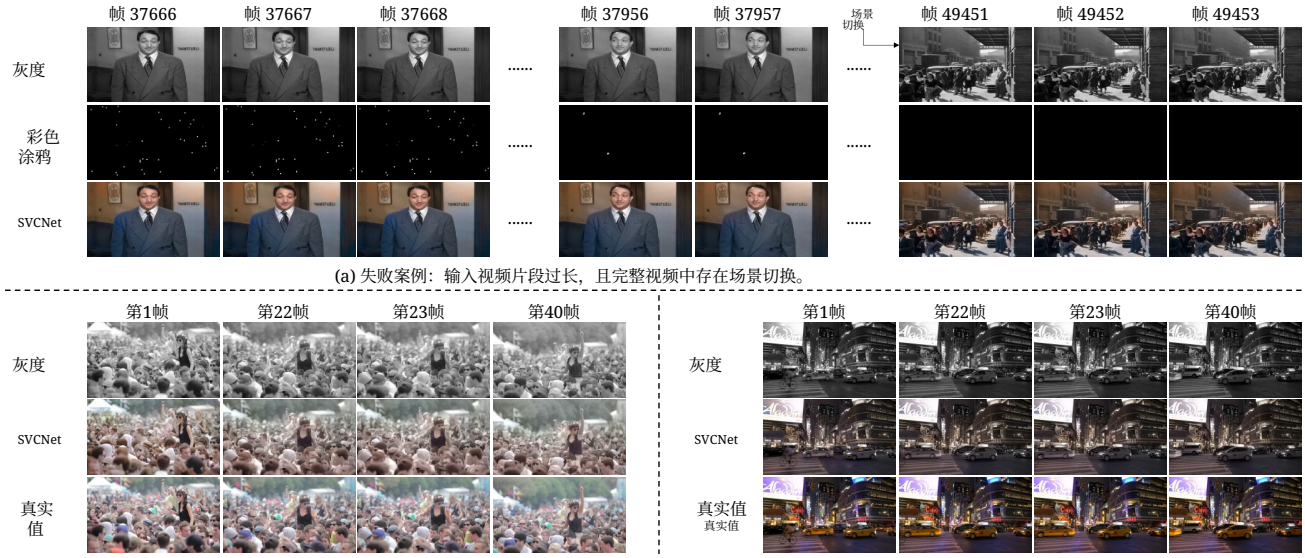
实验设置。 为了进一步展示 SVCNet 的着色质量，我们将 SVCNet 的 CPNet 与以下基线方法进行比较：

1）全自动方法：CIC [52], LTBC [53], LRAC[51],Pix2Pix [54], DeOldify [101],FAVC [9], ChromaGAN[57],SCGAN [59],IAC [58];

2）基于涂鸦的方法：RUIC [30]。

所有方法均在 ImageNet 的 130 万张训练集上训练，并在 ImageNet 的 10000 张验证集上评估，具体定义见 [51], [52], [59]。所有方法均在 256×256 图像分辨率下进行训练和评估。实验中，我们为 SVCNet 和 RUIC 使用了 40 个颜色涂鸦。

定性分析。 定性样例如图10所示。首先，与其他方法相比，CPNet 的结果中颜色渗漏伪影更少（即一个对象的颜色不会渗透到其他对象上）。这是因为我们采用了分割分支和分割损失，帮助网络聚焦于关键对象，并将它们与其他对象区分开来。相比之下，其他方法存在明



(a) 失败案例：输入视频片段过长，且完整视频中存在场景切换。

(b) 失败案例：输入视频片段中的纹理过多。实验使用了 Videvo 数据集中的“Festival”和“TimeSquareTraffic”样本。

图11. 两种常见失败案例的示例：(a) 输入视频片段过长，且发生了场景切换；(b) 纹理过多。
子图(a)中的帧来自一部帧率为24的1948年黑白电影《裸城》。子图(b)省略了输入彩色涂鸦。

表V 图像着色方法与SVCNet的CPNet在ImageNet数据集上的比较。

方法	彩色涂鸦	PSNR	SSIM
CIC [52]	/	22.62	0.9153
LTBC [53]	/	24.96	0.9464
LRAC [51]	/	24.49	0.9229
Pix2Pix [54]	/	23.39	0.9386
DeOldify [101]	/	23.14	0.9194
FAVC [9]	/	22.96	0.9146
ChromaGAN [57]	/	23.67	0.9273
SCGAN [59]	/	23.93	0.9470
IAC [58]	/	24.91	0.9110
VCGAN [11]	/	24.58	0.9427
RUIC [30] （不含 s）	/	25.69	0.9526
SVCNet 的 CPNet（不含 s）	/	25.74	0.9577
RUIC [30] （含 s）	✓	28.94	0.9640
SVCNet 的 CPNet（含 s）	✓	31.40	0.9760

显的伪影，例如 LRAC、CIC 和 LTBC 的“卡车”图像块，以及 RUIC 的“黑猩猩”和“会议”图像块等。其次，即使没有给定颜色涂鸦，CPNet 的结果中颜色混淆问题（即某些对象的颜色在语义上不正确）也更少。例如，对于蜻蜓样例，IAC、DeOldify 和 RUIC（不含 s）将背景着色为蓝色。对于杯子样例，CIC、IAC 和 RUIC（不含 s）着色的背景不一致。LRAC、CIC、LTBC 和 ChromaGAN 对“卡车”的着色也不一致。然而，CPNet 对这些样例的着色更加合理。总之，所提出的 CPNet 具有更强的基于涂鸦的图像着色能力，可作为 SVCNet 的强大骨干网络。

定量分析。定量结果汇总于表 V。一方面，CPNet 的表现优于当前最先进的基于涂鸦的图像着色方法 RUIC，无论是否使用彩色涂鸦（不使用 s）还是使用彩色涂鸦（使用 s）。尤其是在使用彩色涂鸦时，CPNet 的表现大幅优于 RUIC，例如峰值信噪比高出 2.46，结构相似性高出 0.0120。基于性能强大的 CPNet，SVCNet 有望实现高质量的视频着色。另一方面，使用彩色涂鸦时，CPNet 的表现优于现有方法。这是因为 CPNet 能够充分利用彩色涂鸦中的信息来引导着色。与传统架构相比，SVCNet 直接使用预训练的语义特征编码器提取特征，并设有解码器分割分支来预测分割图，这有助于网络处理颜色渗漏伪影。网络设计也有助于提升性能。

G. 失败案例与讨论

SVCNet 可以通过将颜色涂鸦传播到灰度视频中，生成高质量的彩色视频。但在以下情况下，结果往往不够合理：1）视频片段非常长，且完整视频中存在场景切换；2）纹理过多，如图11所示。

首先，我们使用一段真实的老旧视频进行实验，如图 11 (a)所示。当我们为某个场景的第一帧（例如帧 37666）提供彩色涂鸦时，该场景中较远帧（例如帧 37666 和帧 37957）的着色结果在色彩风格上与第一帧保持一致。然而，对较远的帧进行着色时（例如帧 49451，它比帧 37666 晚约 8 分钟），细节表现并不理想。帧 49451 的色彩风格也与第一个着色帧（即帧 37666）非常相似，因为 SVCNet 仅使用长程连接作为参考；然而，两个不同场景的风格并不总是相似。我们认为，应结合关键帧检测算法，为较远的关键帧提供新的彩色涂鸦 [102]。

其次，当灰度视频中包含大量细节时，SVCNet 难以为微小物体着色，如图11 (b)所示。未来，我们将开发更强大的颜色传播技术，并增强 SVCNet 对复杂纹理的鲁棒性。

V. 结论

本文提出了首个基于涂鸦的视频着色框架 SVCNet。该框架由两个顺序连接的子网络组成，分别是颜色传播网络（CPNet）和时空平滑网络（SSNet）。CPNet 根据给定的颜色涂鸦进行精准的图像着色。它利用两个特征编码器有效提取语义信息，并融合颜色涂鸦与灰度图像的信息。此外，它还包含两个解码器分支，一个用于生成颜色嵌入，另一个用于生成相应的分割图。通过在训练期间施加多任务损失，分割分支有助于 CPNet 减轻颜色渗漏伪影。SSNet 对 CPNet 的输出进行后处理，整合短程连接（相邻的着色帧）、长程连接（第一帧着色帧）以及当前时间步的信息（当前 CPNet 输出），以实现良好的时间一致性。此外，我们注意到颜色嵌入具有稀疏性，因此将大多数操作设定在较低的固定分辨率下，并在 SVCNet 的尾部使用超分辨率模块恢复更高分辨率，以满足高清视频应用的需求。最后，我们将 SVCNet 与多种最先进的图像着色和视频着色方法进行了比较。结果表明，与现有方法相比，SVCNet 能够生成更逼真的结果，并且产生的颜色渗漏伪影更少。

参考文献

[1] N. Bonneel、J. Tompkin、K. Sunkavalli、D. Sun、S. Paris 和 H. Pfister，《盲视频时间一致性》，*ACM* 图形学汇刊，第34卷，第6期，第1–9页，2015年。[2] W.-S. Lai、J.-B. Huang、O. Wang、E. Shechtman、E. Yumer 和 M.-H. Yang，《学习盲视频时间一致性》，载于欧洲计算机视觉会议论文集，2018年，第170–185页。[3] Y. Zhou、X. Xu、F. Shen、L. Gao、H. Lu 和 H. T. Shen，《用于学习盲视频时间一致性的时序去噪掩码合成网络》，载于*ACM* 多媒体会议论文集，2020年，第475–483页。[4] C. Lei、Y. Xing 和 Q. Chen，《通过深度视频先验实现盲视频时间一致性》，载于神经信息处理系统大会论文集，2020年，第1083–1093页。[5] V. Jampani、R. Gadde 和 P. V. Gehler，《视频传播网络》，载于*IEEE*计算机视觉与模式识别会议论文集，2017年，第451–461页。[6] S. Meyer、V. Cornill`ere、A. Djelouah、C. Schroers 和 M. Gross，《深度视频颜色传播》，载于英国机器视觉会议论文集，2018年，第85.1–85.10页。[7] R. Endo、Y. Kawai 和 T. Mchizuki，《面向广播节目制作的实用单色视频着色框架》，*IEEE* 广播汇刊，2020年。[8] R. Wu、H. Lin、X. Qi 和 J. Jia，《用于视频传播的记忆选择网络》，载于欧洲计算机视觉会议论文集，2020年，第175–190页。[9] C. Lei 和 Q. Chen，《利用自正则化和多样性实现全自动视频着色》，载于*IEEE*计算机视觉与模式识别会议论文集，2019年，第3753–3761页。[10] P. Kouzouglidis、G. Sfikas 和 C. Nikou，《使用三维条件生成对抗网络进行自动视频着色》，载于智能视觉计算国际会议论文集，2019年，第209–218页。[11] Y. Zhao、L.-M. Po、W.-Y. Yu、Y. A. U. Rehman、M. Liu、Y. Zhang 和 W. Ou，《VCGAN：采用混合生成对抗网络的视频着色》，*IEEE* 多媒体汇刊，第1–1页，2022年。[12] B. Zhang、M. He、J. Liao、P. V. Sander、L. Yuan、A. Bermak 和 D. Chen，《基于深度样例的视频着色》，载于*IEEE*计算机视觉与模式识别会议论文集，2019年，第8052–8061页。[13] S. Iizuka 和 E. Simo-Serra，《Deepremaster：用于全面视频增强的时序源参考注意力网络》，ACM 图形学汇刊，第 38 卷，第 6 期，第 1–13 页，2019年。[14] Z. Wan、B. Zhang、D. Chen 和 J. Liao，《让老电影重获新生》，arXiv 预印本 arXiv:2203.17276，2022年。[15] O. Russakovsky、J. Deng、H. Su、J. Krause、S. Satheesh、S. Ma、Z. Huang、A. Karpathy、A. Khosla、M. Bernstein 等，《ImageNet 大规模视觉识别挑战赛》，《国际计算机视觉杂志》，第 115 卷，第 3 期，第 211–252 页，2015年。[16] C. Liu、J. Yuen 和 A. Torralba，《SIFT Flow：跨场景稠密对应及其应用》，IEEE 模式分析与机器智能汇刊，第 33 卷，第 5 期，第 978–994 页，2010年。[17] J. Chen、A. Adams、N. Wadhwa 和 S. W. Hasinoff，《双边引导上采样》，ACM 图形学汇刊，第 35 卷，第 6 期，第 1–8 页，2016年。[18] S. Guadarrama、R. Dahl、D. Bieber、M. Norouzi、J. Shlens 和 K. Murphy，《Pixcolor：像素递归着色》，载于英国机器视觉会议论文集，2017年，第 112.1–112.13 页。[19] Y. Zhao、Y. Xu、Q. Yan、D. Yang、X. Wang 和 L.-M. Po，《D2HNet：采用层次网络联合去噪与去模糊，实现稳健的夜间图像复原》，载于欧洲计算机视觉会议论文集，2022年。[20] F. Perazzi、J. Pont-Tuset、B. McWilliams、L. Van Gool、M. Gross 和 A. Sorkine-Hornung，《用于视频目标分割的基准数据集与评估方法》，载于IEEE计算机视觉与模式识别会议论文集，2016年，第 724–732 页。[21]《Videvo》，https://www.videvo.net。[22] A. Levin、D. Lischinski 和 Y. Weiss，《基于优化的着色》，ACM 图形学汇刊，第 23 卷，第 3 期，第 689–694 页，2004年。[23] Y.-C. Huang、Y.-S. Tung、J.-C. Chen、S.-W. Wang 和 J.-L. Wu，《一种基于自适应边缘检测的着色算法及其应用》，载于 ACM 多媒体会议论文集，2005年，第 351–354 页。[24] L. Yatziv 和 G. Sapiro，《利用色度混合快速进行图像和视频着色》，IEEE 图像处理汇刊，第 15 卷，第 5 期，第 1120–1129 页，2006年。[25] K. Xu、Y. Li、T. Ju、S.-M. Hu 和 T.-Q. Liu，《利用 k-d树高效进行基于亲和度的编辑传播》，ACM 图形学汇刊，第 28 卷，第 5 期，第 1–6 页，2009年。[26] X. Chen、D. Zou、Q. Zhao 和 P. Tan，《流形保持编辑传播》，ACM 图形学汇刊，第 31 卷，第 6 期，第 1–7 页，2012年。[27] L. Xu、Q. Yan 和 J. Jia，《用于图像和视频编辑的稀疏控制模型》，ACM 图形学汇刊，第 32 卷，第 6 期，第 1–10 页，2013年。[28] B. Sheng、H. Sun、M. Magnor 和 P. Li，《在特征空间中通过并行优化进行视频着色》，IEEE 电路与系统视频技术汇刊，第 24 卷，第 3 期，第 407–417 页，2013年。[29] S. Paul 、 S. Bhattacharya 和 S. Gupta，《利用三维可转向金字塔进行视频时空着色》，IEEE 电路与系统视频技术汇刊，第 27 卷，第 8 期，第 1605–1619 页，2016年。[30] R. Y. Zhang、J. Y. Zhu、P. Isola、X. Geng、A. S. Lin、T. Yu 和 A. A. Efros，《利用学习型深度先验进行实时用户引导图像着色》，ACM 图形学汇刊，第 36 卷，第 4 期，第 119 页，2017年。[31] L. Zhang、C. Li、T.-T. Wong、Y. Ji 和 C. Liu，《两阶段素描着色》，ACM 图形学汇刊，第 37 卷，第 6 期，第 1–14 页，2018年。[32] Y. Ci、X. Ma、Z. Wang、H. Li 和 Z. Luo，《利用条件对抗网络进行用户引导的深度动漫线稿着色》，载于 ACM 多媒体会议论文集，2018年，第 1536–1544 页。[33] Y. Zhao、L.-M. Po、T. Lin、X. Wang、K. Liu、Y. Zhang、W.-Y. Yu、P. Xian 和 J. Xiong，《利用学习型噪声先验进行旧照片编辑》，载于 IEEE/CVF 冬季计算机视觉应用会议论文集，2021年，第 2103–2112 页。[34] E. Reinhard、M. Adhikhmin、B. Gooch 和 P. Shirley，《图像间的颜色迁移》，IEEE 计算机图形学与应用，第 21 卷，第 5 期，第 34–41 页，2001年。[35] T. Welsh、M. Ashikhmin 和 K. Mueller，《向灰度图像迁移颜色》，载于 ACM计算机图形学特别兴趣小组论文集，2002年，第 277–280 页。[36] R. Ironi、D. Cohen-Or 和 D. Lischinski，《基于示例的着色》，载于欧洲图形学渲染研讨会论文集，2005年，第 201–210 页。[37] Y.-W. Tai、J. Jia 和 C.-K. Tang，《通过期望最大化进行概率分割的局部颜色迁移》，载于 IEEE计算机视觉与模式识别会议论文集，第 1 卷，2005年，747 754pp. – 。[38] G. Charpiat、M. Hofmann 和 B. Sch¨olkopf，《通过多模态预测实现自动图像着色》，载于欧洲计算机视觉会议论文集，2008年，第 126–139 页。[39] A. Y.-S. Chia、S. Zhuo、R. K. Gupta、Y.-W. Tai、S.-Y. Cho、P. Tan 和 S. Lin，《利用互联网图像进行语义着色》，ACM 图形学汇刊，第 30 卷，第 6 期，第 1–8 页，2011年。[40] A. Bugeau、V.-T. Ta 和 N. Papadakis，《变分示例图像着色》，IEEE 图像处理汇刊，第 23 卷，第 1 期，第 298–307 页，2013年。

[41] 深度示例着色，ACM 图形学汇刊，第 37 卷，第 4 期，页码 47，2018年。[42] T.-H. Sun、C.-H. Lai、S.-K. Wong 和 Y.-S. Wang，《基于结构和颜色条件的图标对抗式着色》，发表于 ACM 多媒体会议论文集，第 683–691 页，2019年。[43] B. Li、Y.-K. Lai、M. John 和 P. L. Rosin，《利用位置感知跨尺度匹配实现自动示例图像着色》，IEEE 图像处理汇刊，第 28 卷，第 9 期，第 4606–4619 页，2019年。[44] Z. Xu、T. Wang、F. Fang、Y. Sheng 和 G. Zhang，《用于快速深度示例着色的基于风格化的架构》，发表于 IEEE计算机视觉与模式识别会议论文集，2020年，第 9363–9372 页。[45] Y. Bai、C. Dong、Z. Chai、A. Wang、Z. Xu 和 C. Yuan，《用于深度示例着色的语义稀疏着色网络》，发表于 欧洲计算机视觉会议论文集，2022年，第 505–521 页。[46] Z. Sheng、H.-L. Shen、B. Yao 和 H. Zhang，《利用单色图像对进行引导式着色》，IEEE 图像处理汇刊，第 32 卷，第 905–920 页，2023年。[47] H. Bahng、S. Yoo、W. Cho、D. Keetae Park、Z. Wu、X. Ma 和 J. Choo，《用文字着色：通过基于文本的调色板生成引导图像着色》，发表于 欧洲计算机视觉会议论文集，2018年，第 431–447 页。[48] C. Zou、H. Mo、C. Gao、R. Du 和 H. Fu，《场景草图的基于语言的着色》，ACM 图形学汇刊，第 38 卷，第 6 期，第 1–16 页，2019年。[49] Z. Chang、S. Weng、Y. Li、S. Li 和 B. Shi，《L-coder：采用颜色-物体解耦变换器的基于语言的着色》，发表于 欧洲计算机视觉会议论文集，2022年，第 360–375 页。[50] Z. Cheng、Q. Yang 和 B. Sheng，《深度着色》，发表于 国际计算机视觉大会论文集，2015年，第 415–423 页。[51] G. Larsson、M. Maire 和 G. Shakhnarovich，《学习用于自动着色的表征》，发表于 欧洲计算机视觉会议论文集，2016年，第 577–593 页。[52] R. Zhang、P. Isola 和 A. A. Efros，《彩色图像着色》，发表于 欧洲计算机视觉会议论文集，2016年，第 649–666 页。[53] S. Iizuka、E. Simo-Serra 和 H. Ishikawa，《让色彩呈现！：联合端到端学习全局和局部图像先验，实现自动图像着色与同步分类》，ACM 图形学汇刊，第 35 卷，第 4 期，文章 110，2016年。[54] P. Isola、J.-Y. Zhu、T. Zhou 和 A. A. Efros，《采用条件对抗网络的图像到图像转换》，发表于 IEEE计算机视觉与模式识别会议论文集，2017年，第 1125–1134 页。[55] A. Royer、A. Kolesnikov 和 C. H. Lampert，《概率图像着色》，发表于 英国机器视觉会议论文集，2017年，第 85.1–85.12 页。[56] J. Zhao、J. Han、L. Shao 和 C. G. Snoek，《像素化语义着色》，《国际计算机视觉杂志》，第 128 卷，第 4 期，第 818–834 页，2020年。[57] P. Vitoria、L. Raad 和 C. Ballester，《ChromaGAN：基于语义类别分布的对抗式图像着色》，发表于 IEEE/CVF 冬季计算机视觉应用会议论文集，2020年，第 2445–2454 页。[58] J.-W. Su、H.-K. Chu 和 J.-B. Huang，《实例感知图像着色》，发表于 IEEE计算机视觉与模式识别会议论文集，2020年，第 7968–7977 页。[59] Y. Zhao、L.-M. Po、K.-W. Cheung、W.-Y. Yu 和 Y. A. U. Rehman，《SCGAN：显著图引导的生成对抗网络着色》，IEEE 电路与系统视频技术汇刊，第 1–1 页，2020年。[60] M. Kumar、D. Weissenborn 和 N. Kalchbrenner，《着色变换器》，发表于 国际学习表征会议论文集，2021年。 [在线]。地址：https: //openreview.net/forum?id=5NA1PinlGFu [61] Y. Wu、X. Wang、Y. Li、H. Zhang、X. Zhao 和 Y. Shan，《借助生成式颜色先验实现生动多样的图像着色》，发表于 国际计算机视觉大会论文集，2021年，第 14 377–14 386 页。[62] R. Pucci、C. Micheloni 和 N. Martinel，《用于图像着色的图像与物体级协同特征》，发表于 IEEE计算机视觉与模式识别会议研讨会论文集，2021年，第 2160–2169 页。[63] G. Kong、H. Tian、X. Duan 和 H. Long，《结合语义分割的对抗式边缘感知图像着色》，IEEE Access，第 9 卷，第 28 194–28 203 页，2021年。[64] C. Saharia、W. Chan、H. Chang、C. Lee、J. Ho、T. Salimans、D. Fleet 和 M. Norouzi，《Palette：图像到图像扩散模型》，发表于 ACM计算机图形学特别兴趣小组论文集，2022年，第 1–10 页。[65] G. Kim、K. Kang、S. Kim、H. Lee、S. Kim、J. Kim、S.-H. Baek 和 S. Cho，《Bigcolor：利用生成式颜色先验为自然图像着色》，发表于 欧洲计算机视觉会议论文集，2022年，第 350–366 页。[66] Z. Huang、N. Zhao 和 J. Liao，《Unicolor：基于变换器的多模态着色统一框架》，ACM 图形学汇刊，第 41 卷，第 6 期，第 1–16 页，2022年。[67] C. Lei、Y. Xing、H. Ouyang 和 Q. Chen，《用于视频一致性与传播的深度视频先验》，IEEE 模式分析与机器智能汇刊，第 45 卷，第 1 期，第 356–371 页，2022年。

[68] C. Lei、X. Ren、Z. Zhang 和 Q. Chen，“通过神经滤波和有缺陷的图谱实现盲视频去闪烁”，arXiv 预印本 arXiv:2303 08120，2023 年。.[69] M.-M. Cheng、N. J. Mitra、X. Huang、P. H. Torr 和 S.-M. Hu，“基于全局对比度的显著区域检测”，IEEE 模式分析与机器智能汇刊，第 37 卷，第 3 期，第 569–582 页，2014年。[70] C. Yang、L. Zhang、H. Lu、X. Ruan 和 M.-H. Yang，“基于图的流形排序的显著性检测”，载于 IEEE计算机视觉与模式识别会议论文集，2013年，第 3166–3173 页。[71] G. Li 和 Y. Yu，“基于多尺度深度特征的视觉显著性”，载于 IEEE计算机视觉与模式识别会议论文集，2015年，第 5455–5463 页。[72] L. Wang、L. Wang、H. Lu、P. Zhang 和 X. Ruan，“采用循环全卷积网络的显著性检测”，载于 欧洲计算机视觉会议论文集，2016年，第 825–841 页。[73] N. Liu 和 J. Han，“Dhsnet：用于显著目标检测的深度分层显著性网络”，载于 IEEE计算机视觉与模式识别会议论文集，2016年，第 678–686 页。[74] X. Zhang、T. Wang、J. Qi、H. Lu 和 G. Wang，“用于显著目标检测的渐进式注意力引导循环网络”，载于 IEEE计算机视觉与模式识别会议论文集，2018年，第 714–722 页。[75] J.-J. Liu、Q. Hou、M.-M. Cheng、J. Feng 和 J. Jiang，“一种用于实时显著目标检测的简单基于池化的设计”，载于 IEEE计算机视觉与模式识别会议论文集，2019年，第 3917–3926 页。[76] Y.-H. Wu、Y. Liu、L. Zhang、M.-M. Cheng 和 B. Ren，“Edn：通过极度下采样网络进行显著目标检测”，IEEE 图像处理汇刊，2022年。[77] L. Wang、H. Lu、X. Ruan 和 M.-H. Yang，“通过局部估计和全局搜索进行显著性检测的深度网络”，载于 IEEE计算机视觉与模式识别会议论文集，2015年，第 3183–3192 页。[78] T. Zhao 和 X. Wu，“用于显著性检测的金字塔特征注意力网络”，载于 IEEE计算机视觉与模式识别会议论文集，2019年，第 3085–3094 页。[79] L. Zhang、J. Wu、T. Wang、A. Borji、G. Wei 和 H. Lu，“用于显著目标检测的多阶段细化网络”，IEEE 图像处理汇刊，第 29 卷，第 3534–3545 页，2020年。[80] A. Siris、J. Jiao、G. K. Tam、X. Xie 和 R. W. Lau，“场景上下文感知的显著目标检测”，载于 国际计算机视觉大会论文集，2021年，第 4156–4166 页。[81] J. Long、E. Shelhamer 和 T. Darrell，“用于语义分割的全卷积网络”，载于 IEEE计算机视觉与模式识别会议论文集，2015年，第 3431–3440 页。[82] H. Zhao、J. Shi、X. Qi、X. Wang 和 J. Jia，“金字塔场景解析网络”，载于 IEEE计算机视觉与模式识别会议论文集，2017年，第 2881–2890 页。[83] L.-C. Chen、G. Papandreou、I. Kokkinos、K. Murphy 和 A. L. Yuille，“DeepLab：采用深度卷积网络、空洞卷积和全连接条件随机场的语义图像分割”，IEEE 模式分析与机器智能汇刊，第 40 卷，第 4 期，第 834–848 页，2017年。[84] L.-C. Chen、Y. Zhu、G. Papandreou、F. Schroff 和 H. Adam，“采用空洞可分离卷积的编码器-解码器语义图像分割模型”，载于 欧洲计算机视觉会议论文集，2018年，第 801–818 页。[85] M. Yang、K. Yu、C. Zhang、Z. Li 和 K. Yang，“用于街景语义分割的 DenseASPP”，载于 IEEE计算机视觉与模式识别会议论文集，2018年，第 3684–3692 页。[86] K. Liu，“Rm3d：基于传统和学习得到的三维描述符，通过数据高效的方式实现稳健的三维场景解析与语义区域合并”，《国际计算机视觉杂志》，第 1–30 页，2022年。[87] K. Liu、A. Xiao、X. Zhang、S. Lu 和 L. Shao，“Fac：通过前景感知特征对比进行三维表示学习”，arXiv 预印本 arXiv:2303.06388，2023年。[88] K. Liu、Y. Zhao、Q. Nie、Z. Gao 和 B. M. Chen，“通过区域级边界感知和实例判别进行弱监督三维场景分割”，载于 欧洲计算机视觉会议论文集，2022年，第 37–55 页。[89] O. Ronneberger、P. Fischer 和 T. Brox，“U-Net：用于生物医学图像分割的卷积网络”，载于 医学图像计算与计算机辅助手术国际会议论文集，2015年，第 234–241 页。[90] D. Sun、X. Yang、M.-Y. Liu 和 J. Kautz，“PWC-Net：利用金字塔、变形和代价体进行光流估计的卷积神经网络”，载于 IEEE计算机视觉与模式识别会议论文集，2018年，第 8934–8943 页。[91] K. Simonyan 和 A. Zisserman，“用于大规模图像识别的超深卷积网络”，载于 国际学习表征会议论文集，2015年。 [在线]。可访问：https://arxiv.org/abs/ 1409 1556 。X. Glorot 和 Y. Bengio，[92]“理解深度前馈神经网络训练的难点”，载于 人工智能与统计国际会议论文集，2010年，第 249–256 页。[93] D. P. Kingma 和 J. Ba，“Adam：一种随机优化方法”，载于 国际学习表征会议论文集，2015年。 [在线]。可访问：https: //openreview.net/forum?id=8gmWwjFyLj [94] K. He、X. Zhang、S. Ren 和 J. Sun，“用于图像识别的深度残差学习”，载于 IEEE计算机视觉与模式识别会议论文集，2016年，第 770–778 页。[95] X. Wang、K. Yu、S. Wu、J. Gu、Y. Liu、C. Dong、Y. Qiao 和 C. Change Loy，“Esrgan：增强型超分辨率生成对抗网络”，收录于*ECCVW* 论文集，2018，第 0–0 页。

[96] A. L. Maas、A. Y. Hannun 和 A. Y. Ng，《整流非线性改进了神经网络声学模型》，载于 国际机器学习大会论文集，第 30 卷，第 1 期，2013年，第 3 页。[97] D. Ulyanov、A. Vedaldi 和 V. Lempitsky，《实例归一化：快速风格化所缺失的关键》，*arXiv* 预印本 *arXiv:1607.08022*，2016年。[98] Z. Wang、A. C. Bovik、H. R. Sheikh、E. P. Simoncelli 等，《图像质量评估：从误差可见性到结构相似性》，*IEEE* 图像处理汇刊，第 13 卷，第 4 期，第 600–612 页，2004年。[99] K. Sun、B. Xiao、D. Liu 和 J. Wang，《面向人体姿态估计的深度高分辨率表征学习》，载于 *IEEE*计算机视觉与模式识别会议论文集，2019年。[100] Y. Yuan、X. Chen、X. Chen 和 J. Wang，《分割变换器：用于语义分割的对象上下文表征》，*arXiv* 预印本 *arXiv:1909.11065*，2019年。[101]《DeOldify》，https://github.com/jantic/DeOldify。[102] X. Yan、S. Z. Gilani、H. Qin、M. Feng、L. Zhang 和 A. Mian，《人体动作视频中的深度关键帧检测》，*arXiv* 预印本 *arXiv:1804.10021*，2018年。



Yuzhi Zhao （S'19）于2018年获得华中科技大学（HUST）电子与信息工程学士学位，并于2023年获得香港城市大学（CityU）电子工程博士学位。其研究兴趣包括低层视觉、计算摄影、生成模型和表征学习。他担任CVPR、ICCV、ECCV、WACV、亚洲计算机视觉会议、IEEE 声学、语音与信号处理国际会议、IEEE 国际图像处理会议等国际会议及多种 IEEE 汇刊的同行评审人。



Lai-Man Po （M'92–SM'09）分别于1988年和1991年获得香港香港城市大学电子工程学士学位和博士学位。自1991年起，他一直任职于香港城市大学电子工程系，目前担任电气工程系副教授。他已发表150多篇期刊和会议论文。其研究兴趣包括图像与视频编码，尤其关注基于深度学习的计算机视觉算法。

Po博士是多媒体系统与应用技术委员会和IEEE电路与系统学会的成员。他于2012年和2013年担任IEEE信号处理学会香港分会主席。2011年至2013年，他担任《香港工程师学会会刊》副编辑。他还曾担任2003年IEEE声学、语音与信号处理国际会议和2010年IEEE图像处理国际会议的组委会成员。



Kangcheng Liu （M'19）目前是新加坡南洋理工大学（NTU）的研究员。他于2018年获得哈尔滨工业大学（HIT）工学学士学位，并于2022年获得香港中文大学（CUHK）机械与自动化工程博士学位，研究方向为机器人学。他曾担任IEEE机器人与自动化国际会议的程序委员会委员，并为IEEE计算机视觉与模式识别会议、国际计算机视觉大会、欧洲计算机视觉会议、ACM计算机图形学特别兴趣小组、国际计算机视觉期刊、IEEE机器人学汇刊和IEEE模式分析与机器智能汇刊审稿。



Xuehui Wang （S'21）分别于2018年和2021年获得中国山东大学理学学士学位和中国中山大学计算机科学学院硕士学位。目前，他正在中国上海交通大学人工智能研究院攻读博士学位。他的研究兴趣包括计算机视觉（超分辨率、实例分割）和深度学习。



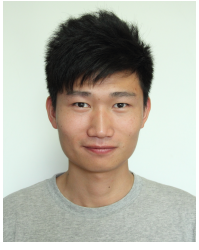
Wing-Yin Yu （S'21）于2019年获得香港城市大学信息工程学士学位。他目前正在香港城市大学电子工程系攻读博士学位。他的研究兴趣包括深度学习和计算机视觉。



Pengfei Xian （S'21）于2017年获得哈尔滨工业大学（中国哈尔滨）电气工程学士学位。他目前正在香港城市大学攻读电气工程博士学位。他的研究兴趣包括图像和视频中的实例分割与语义分割，以及强化学习应用。



Yujia Zhang 于2015年在华中科技大学获得电气工程及其自动化工学学士学位，于2018年在华南理工大学获得电气工程理学硕士学位，并于2022年在香港城市大学获得电气工程博士学位。他目前是腾讯控股有限公司旗下腾讯视频的一名工程师。其当前研究兴趣包括计算机视觉和视频理解。



Mengyang Liu 于2014年在中国上海的上海电力大学获得光电信息工程工学学士学位，并于2015年和2019年分别在香港城市大学获得电子与信息工程理学硕士学位和博士学位。他目前是腾讯控股有限公司旗下腾讯视频的一名工程师。其研究兴趣包括图像与视频处理、视频嵌入与检索、计算机视觉和机器学习。