# Cross-CNN：基于CNN和Transformer混合模型的动画跨帧线稿着色算法

余毅丰[1,2]，钱江波[1,2*]，严迪群[1,2]，王 翀[1,2]，董 理[1,2]
（1. 宁波大学信息科学与工程学院，浙江宁波 315000；2. 浙江省移动网应用技术重点实验室，浙江宁波 315000）

**摘 要：** 对长序列的动画线稿帧进行着色是计算机视觉中一项具有挑战性的任务. 一方面，线稿中包含的信息较为稀疏，需要着色算法对缺失的信息进行推断；另一方面，连续帧之间的色彩需要保持一致，以确保整个视频的视觉质量. 现有的着色算法多数只针对单张图片进行着色，这类算法只给出一个开放性的符合合理范围的色彩结果，无法适用于帧序列着色. 另一些基于参考帧的着色算法，并没有将2帧之间的关系有机地联系起来，导致着色效果不够出色. 在同一镜头序列中，同一对象的特征往往不会发生太大变化，因此，可以设计一个根据给定参考帧，即可给线稿自动着色的模型. 为此，本文提出了基于CNN（Convolutional Neural Networks）和Transformer相结合的模型Cross-CNN，该模型能够从参考帧中寻找并匹配颜色，从而保证时间维度上的特征一致性. Cross-CNN模型参考帧和线稿帧在通道维度叠加，输入预训练的Resnet50网络提取局部融合特征，将融合特征图传给Transformer结构进行编码以提取全局特征. 在Transformer结构中设计了交叉注意力机制更好地匹配远距离特征. 最后使用带有跳层连接的卷积解码器完成着色图片输出. 本文在数据集方面从8部电影中截取画面并经过严格筛选，最终制作了一个包含20 000对二元组的数据集用于实验研究. Cross-CNN的SSIM（Structural SIMilarity）达到了0.932，高于SOTA算法0.014. 本文算法代码链接：https://github.com/silenye/Cross-CNN.

**关键词：** 线稿着色；卷积神经网络；Transformer；颜色匹配；动画制作
**基金项目：** 国家自然科学基金（No.62271274）；宁波市科技项目（No.2024Z004，No.2023Z059）
**中图分类号：** TP391.41 **文献标识码：** A **文章编号：** 0372-2112(2024)07-2491-12
**电子学报URL:** http://www.ejournal.org.cn **DOI:** 10.12263/DZXB.20230622

# Cross-CNN: An Animation Cross-Frame Sketch Colorization Algorithm Based on Hybrid Model with CNN and Transformer

YU Yi-feng[1,2], QIAN Jiang-bo[1,2*], YAN Di-qun[1,2], WANG Chong[1,2], DONG Li[1,2]
（1. *Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, Zhejiang* 315000, *China*;
2. *Zhejiang Key Laboratory of Mobile Network Application Technology, Ningbo, Zhejiang* 315000, *China*）

**Abstract:** Coloring long sequences of animated sketch frames is a challenging task in computer vision. On one hand, the information contained in sketches is sparse, and coloring algorithms need to infer missing information. On the other hand, the colors between consecutive frames need to be consistent to ensure visual quality throughout the video. Most existing coloring algorithms are designed for single images and only provide one open-ended, reasonable color result, which is not suitable for coloring frame sequences. Other reference-based coloring algorithms do not have an organic connection between two frames, resulting in unsatisfactory coloring results. In the same shot sequence, the features of same object usually do not change too much. Therefore, a model that can automatically color sketches based on a given reference frame can be designed. This paper proposes a new model called Cross-CNN that combines convolutional neural networks (CNN) and Transformer. Our Cross-CNN can find and match colors from the reference frame, thus ensuring temporal feature consistency. In this model, the reference frame and the sketch frame are superimposed in the channel dimension, and the pre-trained Resnet50 network is used to extract locally fused features. The fused feature map is then passed to the Transformer structure for encoding to extract global features. In the Transformer structure, a cross attention mechanism is designed to better match

收稿日期：2023-07-03；修回日期：2023-10-25；责任编辑：王天慧
*通讯作者：钱江波

long-distance features. Finally, a convolutional decoder with skip connections is used to output the colored image. In terms of the dataset, this paper extracted frames from eight movies and conducted strict screening to create a dataset containing 20 000 pairs of reference and sketch frames for experimental research. The SSIM (Structural SIMilarity) of Cross-CNN can reach 0.932, which is higher than the SOTA algorithm by 0.014. The algorithm codes link for this paper: https://github.com/silenye/Cross-CNN.

Key words: sketch coloring; convolutional neural network; Transformer; color matching; animation production

Foundation Item(s): National Natural Science Foundation of China (No.62271274); Ningbo Science and Technology Project (No.2024Z004, No.2023Z059)

## 1 引言

动画制作可以简单概括为几个步骤:(1)确定故事情节;(2)设计相应的人物角色;(3)设计叙事镜头;(4)动画原画制作;(5)动画分镜设计;(6)动画帧着色;(7)音乐和配乐制作;(8)后期制作[1]. 其中,除了步骤(6)外,其他工序都需要根据人的创意及想法来主导,无法将此类工序交由计算机完成. 而步骤(6)中,高级画师定义主要角色的关键帧线稿,普通画师通过关键帧扩充中间的线稿,根据主创设计的人物色卡,对所有线稿进行着色[2]. 虽然随着计算机技术的发展,出现了手绘板等硬件设备以及各种软件用于动画制作,但对线稿进行着色仍是1个高度手动的过程,需要绘制和处理每1帧[3~5]. 这一复杂且劳动密集的过程导致动画制作的资金和时间成本居高不下[6]. 而计算机的优势在于强大的计算能力可以用于重复型任务. 因此,计算机可以取代人工来完成繁琐反复的线稿着色任务,高级画师画完关键帧之后,可将关键帧交由计算机,使其自动分析绘画风格,并替代人工对剩余线稿进行着色. 随着深度学习的发展,使得计算机自动对线稿进行着色成为可能. 深度学习用于线稿着色的好处有以下几点:

(1)提高着色的效率:与传统的人工着色相比,基于深度学习的方法能够在更短时间内完成着色,从而提高制作效率,节约成本.

(2)提高着色的可控性:深度学习通过学习大量数据,可为用户提供更丰富的创意编辑,例如,通过调整模型参数、引入不同约束实现个性化着色需求.

(3)易于扩展性:与人工相比,模型的学习能力更快. 只要增加数据量,模型即可掌握越来越多的绘画风格,应用的范围会变广. 而人工很难在短期内像模型一样精通各种绘画风格.

近年来,对图像进行着色取得了长足进步. 例如对灰度图、简笔人物进行着色. 然而,并没有合适的方法对动画图片进行着色. 这是由于与单一图像相比,动画往往由多帧组成,要求每1帧的色彩风格在时间序列中是统一的,同一物体的颜色特征在一个镜头中往往不会有明显变化. 其他基于参考帧的算法,虽然学习了参考帧中色彩的分布特征来给线稿着色,但着色结果仍然会与真实色彩分布存在偏差. 为解决参考帧与线稿间隔较远导致着色效果不佳的问题,本文提出了基于卷积神经网络(Convolutional Neural Networks,CNN)和Transformer的混合着色模型Cross-CNN. 间隔较远的帧导致2帧之间画面相似度低,因此,希望尽可能利用参考帧中的色彩信息. 浅层CNN优势在于提取局部特征,Transformer优势在于提取全局特征. 因此,模型利用浅层CNN来提取局部的色彩特征,之后传入Transformer来提取全局特征,增强模型对参考帧中信息提取的能力. 具体来说,先输入参考帧和线稿帧至预训练Resnet50网络,然后提取不同深度的3层融合特征,用于解码器阶段的跳层连接. 同时,取上述3层特征中最深的1层融合特征,传入Transformer模块,进行注意力计算,Transformer模块中将图片变为token,以patch的形式进行精细的特征对比. 考虑到参考帧与线稿帧距离间隔较远,而传统的注意力机制产生的$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$矩阵来自于1张输入帧,所以,本文提出了交叉注意力机制,计算注意力的时候,$\boldsymbol{K}$矩阵由参考帧产生,$\boldsymbol{Q}$、$\boldsymbol{V}$矩阵由混合帧产生,$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$矩阵进行注意力计算,得到参考帧和线稿帧的混合特征. 计算注意力时产生的$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$矩阵来自于不同的2帧,因此,计算注意力的时候能够同时计算参考帧与线稿帧的特征,产生更好着色结果. 参考帧的使用能够输出更接近用户想法的结果,而使用交叉注意力能使输出的结果更接近于参考帧中的色域. 针对数据集缺少问题,本文自行收集了动画数据集用于研究. 该方法的主要贡献可以总结如下:

(1)使用CNN和Transformer的混合模型Cross-CNN,通过设置参考帧和线稿帧,完成跨帧的线稿着色.

(2)提出交叉注意力机制,使patch之间的计算能充分融合2帧输入之间的特征,产生更接近于参考帧色域的结果.

(3)针对缺乏相关数据集的问题,本文从多部电影中采样五元组来制作数据集. 其中,参考帧和线稿帧之间的距离存在多种可能,数据集也包含各种场景.

## 2　相关工作

根据着色的应用类型,可以将着色工作分为黑白图片和线稿着色.根据是否有颜色约束可以分为无参考帧的着色与基于参考帧或人工提示的着色方法.

### 2.1　黑白图像着色

早期的工作聚焦于利用非深度学习的方法来对灰度图着色.Horiuchi等[7]提出一种彩色种子传播算法,从候选颜色列表中选择颜色来完成灰度图的着色.但是最终的结果会产生失真伪影(visible artifacts).Levin等[8]基于时空中具有相似强度的相邻像素应该具有相似颜色(neighboring pixels in space-time that have similar intensities should have similar colors)的假设,提出了1种简单着色方法,无需精确分割以及区域追踪,只需要一些颜色涂鸦用于提示便可给灰度图进行着色.Qu等[9]利用基于纹理的水平集方法(texture-based level set method),能够在相似但不一定均匀的区域上传播边界曲线,用于漫画的彩色化.Sýkora等[10]将着色问题重新表述为关于泊松相互作用(Potts interaction)和特殊稀疏数据项(special sparse data term)的能量优化变体问题,并提出了1种有效的近似算法.Yatziv等[11]从亮度通道(luminance channel)计算加权距离函数用于颜色混合,但仍需用户给出颜色信息提示.这些算法虽然能产生一些令人印象深刻的着色,但着色准确度取决于用户提示,对于复杂图像,需要用户给予密集的颜色信息,需要大量人力工作.

### 2.2　无参考帧着色

在图像生成领域,GAN[12]由于其卓越的效果而备受青睐.Pix2Pix[13]使用cGAN[14],寻找输入图像像素到输出图像像素的映射,但这种一对一映射的应用范围有限.CycleGAN[15]通过2个生成器和2个判别器,来学习2个数据分布之间的转换映射.Nazeri等[16]通过修改损失函数,将GAN用于着色任务,借鉴了cGAN的思想,允许灰度图作为生成器的输入,取代了随机噪声的不确定性.Su等[17]提出了实例感知着色模型,用于灰度图着色.模型通过第三方预训练检测模型,先提取出局部对象,分别对局部对象和背景进行着色,通过融合模块对2部分着色结果进行融合.但在着色之前,需要提取物体信息,导致部分无主体图片无法进行很好着色.Yoo等[4]提出1种记忆增强网络用于灰度图着色.不同于别的网络直接从图像中学习色彩特征,该网络开辟了外部内存,专门存储从当前对象中提取的有用色彩信息.在训练阶段,网络通过不断学习,更新内存中最匹配当前物体特征的颜色信息.在使用阶段,网络查询外部内存,提取出最匹配的颜色信息给输入的灰度图着色.该网络有效避免了主色效应,保留不同物体的颜色特征.金正猛等[18]在YCbCr色彩空间机制下,结合目标灰度图像的梯度信息,提出基于耦合全变差的图像着色模型.利用交替方向乘子算法设计模型的快速数值求解,给出该算法的收敛性结果.李洪安等[19]提出了结合Pix2Pix的灰度图着色算法,该算法加深了U-Net结构[20],使用$L_1$损失和smooth$L_1$损失来度量生成图像和真实图像之间的差距,并对每个输入数据进行梯度惩罚.但以上基于图像翻译任务产生的结果不受使用者控制,输出符合颜色语义的结果,例如汽车可以是红色也可以是黑色,无法准确得到用户想要的精确结果.参考帧的缺失可能会使着色结果不尽如人意.

### 2.3　基于参考帧或人工提示着色

区别于传统的无参考帧着色的GAN,一些算法通过用户线索或参考帧对线稿或灰度图像进行着色.Zhang等[21]提出的模型给定1帧彩色图片和1帧线稿,通过VGG(Visual Geometry Group)网络提取彩色图片的色彩风格特征,作为参考信息给线稿着色.Zhang等[22]提出了两阶段着色方法,第一阶段给定部分像素色彩,第二阶段网络通过像素色彩信息对图片进行精细化着色.但不能使用整版图片作为参考,用户也需要耗费大量时间来给定像素色彩信息.Zhang等[23]提出的模型通过语义特征检索接近参考源,并利用语义相关性对灰度图进行着色.虽然这种方法在照片着色方面有良好表现,但高度抽象稀疏的线稿中并没有提供密集的颜色线索.Liu等[24]使用颜色样式提取器从彩色参考图片中提取色彩特征,并与线稿融合来完成着色.此方法没有利用时间信息,如果用于序列帧着色,可能导致颜色不一致.Lee等[25]提出的模型使用参考帧给草图着色.对于1张彩色图,先使用轮廓提取器获取图片线稿,然后通过色彩变化和空间变化,得到参考帧和标签帧.参考帧可以提供准确的颜色信息,空间变化是为了区别线稿与参考帧的纹理特征.Zhang等[26]设计了1种相关匹配特征传输模型,以可学习的方式对齐参考彩色特征,并以从粗到细的方式将该模型集成到基于U-Net的生成器中.这使得生成器能将分层同步特征从深层语义逐步转移到具体内容.UGSC-GAN[27]设计了1种新颖的草图着色方法,旨在从二元草图和稀疏色域重建动漫照片.该框架由2个阶段组成,第一阶段从生成器中生成灰色图像,旨在重建边缘和纹理信息.第二阶段参照用户提示或彩色图像,对第一阶段生成的灰色图像进行着色.Cho等[28]提出了GuidingPainter模型用于线稿的交互式着色,模型能按照优先级主动寻找需要提供颜色的区域,引导用户给予色彩提示,显著提高交互式着色的效率.Li等[29]提出1种新的视频合成方法,该方法使用2个颜色帧来获得颜色信息,并能根

据输入的线稿引导网络输出更接近用户意图的结果. Shi 等[3]设计了2个网络,1个用于着色,另1个3D网络用于监测结果的时间一致性,输入1个或多个参考帧来完成线稿的着色. Hensman 等[30]提出的模型接受1张参考帧用于动漫的着色. 利用cGAN完成初步的着色,还利用色彩分割以及颜色校正等方法来输出锐利清晰的结果,但对于草图的着色不够理想. Thasarathan 等[31]提出的模型也采用类似cGAN的模型,用于灰度图或者线稿着色,但没有考虑时间维度的一致性. Li 等[32]提出了1种新颖的注意力机制SGA(Stop-Gradient Attention),该机制通过保留主要梯度分支,同时,去除冲突分支来确保不精确梯度与精确梯度具有正余弦相似度,缓解梯度问题. Lin 等[33]提出1个家庭场景图像色彩迁移的框架. 在将图像划分为局部区域并提取其对应的颜色后,根据原始家庭场景图像的颜色结构对模板图像进行采样,生成1个匹配色表,最后,在保持边界转换的情况下,将颜色从匹配色表转换到目标家庭场景图像. Liu 等[34]和 Vondrick 等[35]提出的模型利用参考帧对灰度视频进行着色,匹配相应的颜色信息,但对于高度稀疏的线稿,包含较少语义特征,这对模型来说是困难的. Cheng 等[36]给定参考帧相较于无参考着色,进一步明确了色彩空间范围,限定了输出的色彩域,用户可以通过需求,设置参考帧,得到更符合用户要求的结果. 图1展示了有无参考帧模型的区别,虚线上面为无参考帧模型,模型直接从线稿抽取特征进行编码并进行上色,虚线下面的有参考帧模型可以分别从彩色图片及黑白线稿提取特征.

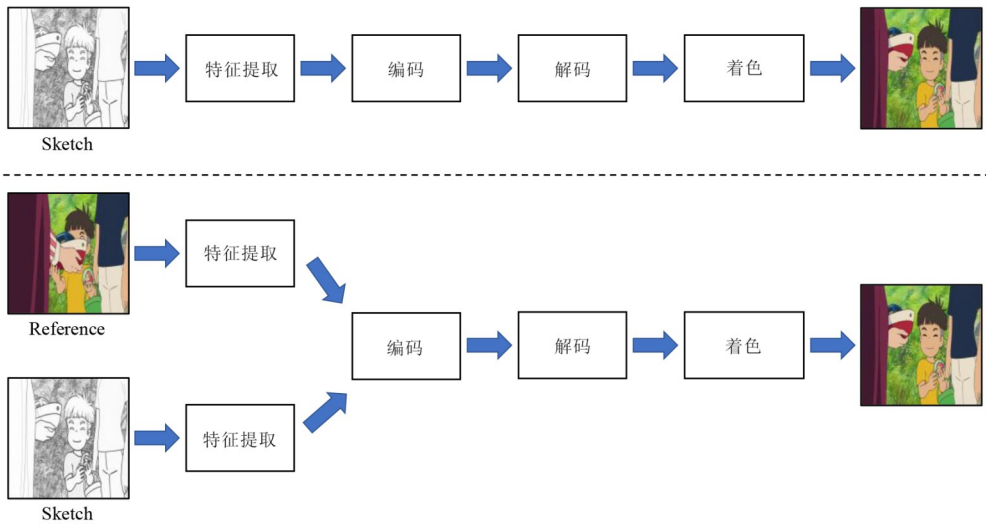



图1 有无参考帧模型对比

## 3 Cross-CNN模型

受Transunet[37]的启发,本文使用CNN和Transform混合网络Cross-CNN来完成着色任务. 网络结构如图2所示.

网络由CNN和Transformer结合的编码器、交叉注意力机制以及U-Net结构的跳层解码器组成,在接下去的小节中分别详细介绍了3部分.

### 3.1 编码器

考虑到线稿上完色输出的图片的色域与参考帧的色域应该一致,所以网络的输入数据由参考帧$\boldsymbol{I}_{\text{reference}}(\boldsymbol{I}_{\text{r}})$和线稿$\boldsymbol{I}_{\text{sketch}}(\boldsymbol{I}_{\text{s}})$组成. 首先将2张图片在通道维度上叠加起来,叠加结果记为$\boldsymbol{x}\in\mathbb{R}^{H\times W\times C}$,$H$、$W$分别表示图片的宽和高,$C$表示图片的通道,此时$C$为6. 其次,将$\boldsymbol{x}$传入训练好的Resnet50网络提取图像的高级特征,因为$\boldsymbol{x}$中同时包含了参考帧$\boldsymbol{I}_{\text{r}}$和线稿帧$\boldsymbol{I}_{\text{s}}$,所以$\boldsymbol{I}_{\text{r}}$中的颜色能与$\boldsymbol{I}_{\text{s}}$融合,提取出融合特征. 卷积因为感受野的缘故,提取的特征都是基于局部特征,只有经过多层网络后,网络才能提取到大感受野特征. 最终得到Resnet50的第3、4、9层block的输出,第3、4、9层block输出保留作为后续解码器阶段跳层连接所需的浅层网络特征,同时,第9层block输出作为后续Transformer编码器的输入. 因为使用了CNN网络提取图像特征,卷积核所提取的是$\boldsymbol{I}_{\text{r}}$和$\boldsymbol{I}_{\text{s}}$的融合特征,所以传入Transformer编码器的是高级特征而非原图,这有利于网络对$\boldsymbol{I}_{\text{s}}$完成初步色彩识别. 由于Transformer接受的是序列化特征,所以需要对CNN提取的高级特征进行处理. 将patch的高和宽都设置为$P=16$,创建1个相同大小的卷积核,以$s=P$的步长对高级特征进行卷积以完成图片序列化. 经过卷积之后,将得到2D的patch集合记为$\left\{\boldsymbol{x}_{\text{p}}^{i}\in\mathbb{R}^{P^2\times C}\mid i=1,2,\cdots,N\right\}$,其中,patch的分辨率为$P\times P$,$N=\dfrac{HW}{P^2}$表示patch的个数. 在网络编码过程中,位置信

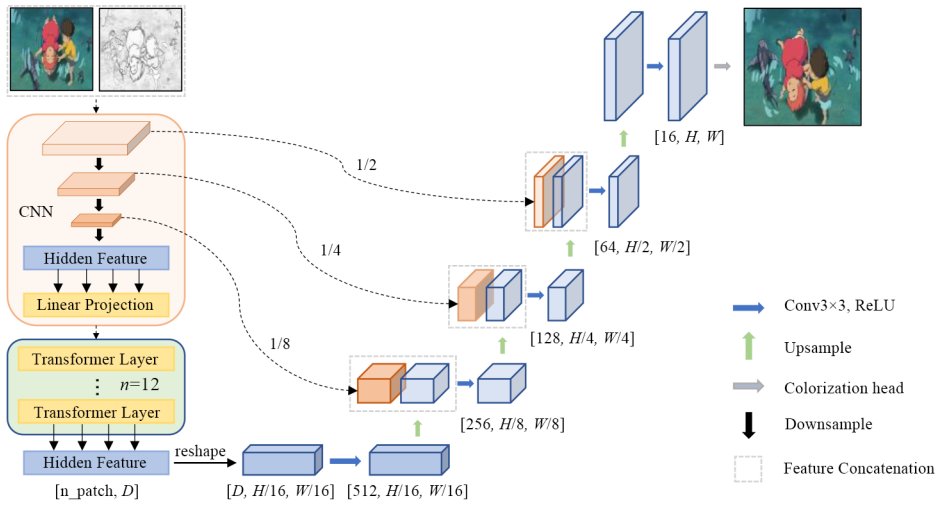



图2　Cross-CNN模型结构

息对于着色任务非常重要,因此,当$\boldsymbol{x}_{\mathrm{p}}$映射到$D$维隐藏embedding层的时候添加位置信息,使网络能够更好学习纹理特征及其对应的色彩特征. 添加的位置信息如以下公式:

$$\boldsymbol{z}_0=\left[\boldsymbol{x}_{\mathrm{p}}^1\boldsymbol{E};\ \boldsymbol{x}_{\mathrm{p}}^2\boldsymbol{E};\cdots;\ \boldsymbol{x}_{\mathrm{p}}^N\boldsymbol{E}\right]+\boldsymbol{E}_{\mathrm{pos}} \tag{1}$$

其中,$\boldsymbol{E}\in\mathbb{R}^{(P^2\times C)\times D}$表示卷积核大小为1×1的embedding投影层;$\boldsymbol{E}_{\mathrm{pos}}\in\mathbb{R}^{N\times D}$表示位置embedding.

在Transformer编码器中设置了12层多头自注意力(Multihead Self-Attention,MSA)以及相对应多层感知机(Multi-Layer Perceptron,MLP)用于提取全局特征. 第$\ell$层的输出如以下公式所示:

$$\boldsymbol{z}'_{\ell}=\mathrm{MSA}\left(\mathrm{LN}\left(\boldsymbol{z}_{\ell-1}\right)\right)+\boldsymbol{z}_{\ell-1} \tag{2}$$

$$\boldsymbol{z}_{\ell}=\mathrm{MSA}\left(\mathrm{LN}\left(\boldsymbol{z}'_{\ell}\right)\right)+\boldsymbol{z}'_{\ell} \tag{3}$$

其中,LN(·)表示Layer Norm,Transformer的结构如图3所示.

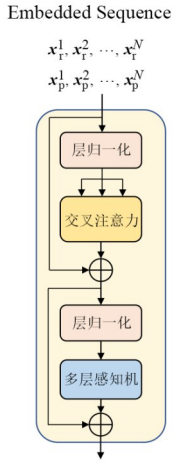



图3　Transformer结构

## 3.2　交叉注意力

本文根据任务和输入,设计了交叉注意力机制(cross attention). 在传统的注意力计算中,$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$矩阵由1个输入产生,再进行下一步计算. 在本文的任务中,由于输入了$\boldsymbol{I}_{\mathrm{r}}$和$\boldsymbol{I}_{\mathrm{s}}$,因此可以选择1张图片来产生$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$其中1个矩阵. 网络需要根据纹理特征来给稀疏的$\boldsymbol{I}_{\mathrm{s}}$填充色彩,所以使用$\boldsymbol{I}_{\mathrm{r}}$来产生$\boldsymbol{K}$矩阵,用于包含$\boldsymbol{I}_{\mathrm{s}}$的混合特征产生的$\boldsymbol{Q}$矩阵来查询对应颜色,同时,$\boldsymbol{I}_{\mathrm{r}}$中包含了纹理特征,也能辅助网络根据纹理特征的变化来学习颜色变化. 此外,因为$\boldsymbol{I}_{\mathrm{r}}$与$\boldsymbol{I}_{\mathrm{s}}$之间间隔较远,交叉注意力设计也有利于网络更细致比较纹理特征的变化,更好地学习色彩的匹配. 图4展示了交叉注意力的示意图,可以看到$\boldsymbol{I}_{\mathrm{s}}$中人物的左手手臂在最中间的patch,而$\boldsymbol{I}_{\mathrm{r}}$中该手臂位于4个不同的patch,将$\boldsymbol{I}_{\mathrm{r}}$中的彩色像素作为$\boldsymbol{K}$,能够使$\boldsymbol{I}_{\mathrm{s}}$产生更接近于$\boldsymbol{I}_{\mathrm{r}}$的颜色. 交叉注意力将2帧有机联系起来,最终达到更好的效果.

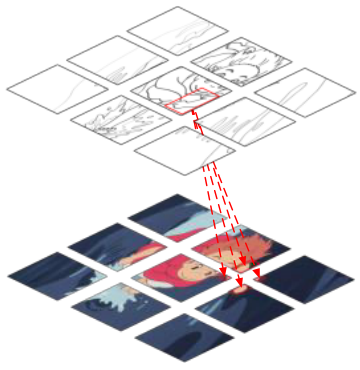


图4　交叉注意力示意图

计算交叉注意力时,需要通过卷积将$\boldsymbol{I}_{\mathrm{r}}$变成序列集合. 首先,将卷积核大小设置为$k=16$,再以$s=k$的步长对$\boldsymbol{I}_{\mathrm{r}}$进行卷积运算,最终得到patch集合

$\left\{\boldsymbol{x}_{\mathrm{r}}^{i}\in\mathbb{R}^{P^2\cdot C}\middle|i=1,2,\cdots,N\right\}$，其中，$N=\dfrac{HW}{P^2}$表示 patch 的个数．该序列集合用于投影层并通过 Layer Norm 得到新的集合$\left\{\mathrm{LN}\left(\boldsymbol{x}_{\mathrm{r}}^{i}\in\mathbb{R}^{P^2\cdot C}\right)\middle|i=1,2,\cdots,N\right\}$来产生$\boldsymbol{K}$矩阵，交叉注意力的计算过程如图 5 所示．同理，$\boldsymbol{Q}$、$\boldsymbol{V}$矩阵由 Resnet50 产生的混合特征经过卷积序列化并通过投影层和 Layer Norm 产生．在第一个交叉注意力模块中输入了$\boldsymbol{I}_{\mathrm{r}}$的特征以及$\boldsymbol{I}_{\mathrm{s}}$的特征，$\boldsymbol{I}_{\mathrm{r}}$的特征和$\boldsymbol{I}_{\mathrm{s}}$的特征进行注意力计算，产生的融合特征传入下一个交叉注意力模块，除第 1 个交叉注意力模块中，剩下的 11 个交叉注意力模块输入上一个模块输出的融合特征以及序列化的$\boldsymbol{I}_{\mathrm{r}}$特征．

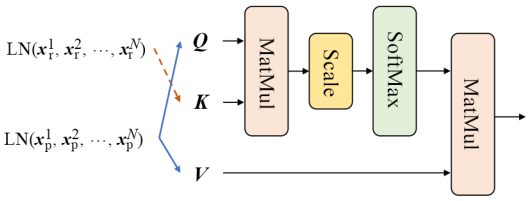



图 5 Cross Attention 计算

Cross Attention 的计算公式如下所示：

$$\mathrm{Attention}\left(\boldsymbol{Q}_{\mathrm{p}},\boldsymbol{K}_{\mathrm{r}},\boldsymbol{V}_{\mathrm{p}}\right)=\mathrm{Softmax}\left(\frac{\boldsymbol{Q}_{\mathrm{p}}\boldsymbol{K}_{\mathrm{r}}^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V}_{\mathrm{p}}\tag{4}$$

为了区分$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$的来源，分别将其记为$\boldsymbol{Q}_{\mathrm{p}}$、$\boldsymbol{K}_{\mathrm{r}}$、$\boldsymbol{V}_{\mathrm{p}}$，表明该矩阵来源于$\boldsymbol{I}_{\mathrm{r}}$或混合特征．其中$\sqrt{d_k}$为缩放因子以避免点积操作带来的方差影响．

### 3.3 解码器

在解码器阶段，网络通过反卷积来重建 1/8、1/4、1/2 以及原始大小的图像．同时，在 1/8、1/4、1/2 图像尺度下，在通道维度上拼接 CNN 编码器阶段通过 Resnet50 提取的特征．反卷积能使网络恢复局部空间信息，跳层连接的应用能使网络有效结合浅层特征以及深层特征，有助于恢复下采样过程中所产生的信息损失．

在网络最后的输出阶段，设置 1 层卷积层来输出 3 通道的图像作为网络的最终输出．此时网络完成着色．

### 3.4 损失函数

在网络训练阶段，使用$L_1$像素损失来监督网络的输出．该损失函数的计算方法是将模型预测出的图像与真实标签图像逐像素比较，计算它们的绝对差值（即每个像素点的差值取绝对值）．使用的$L_1$损失函数如下：

$$L_1=\left\|\boldsymbol{y}-\hat{\boldsymbol{y}}\right\|_1\tag{5}$$

其中，$\boldsymbol{y}$表示网络输出的图片；$\hat{\boldsymbol{y}}$表示标签图片．相对于$L_2$损失，$L_1$损失对异常值更加敏感，因为使用了绝对值，而不是平方，因此，它能够更好地保留图像细节和纹理．

## 4 实验

在本节中，详细介绍了收集数据集的过程、学习策略、评价指标以及对比模型．在数据集上进行广泛实验以证明本文算法的有效性．

### 4.1 数据集

本文实验所使用的数据来自于 8 部电影（Howl's Moving Castle、Karigurashi no Arrietty、Laputa Castle in the Sky、Mononoke Hime、My Neighbour Totoro、Ponyo On The Cliff by The Sea、Spirited Away、The Wind Rises）．经过测试，设置 8 帧作为间隔从原始电影片段中采样 5 帧组成五元组．以 8 帧作为间隔能够在确保间隔距离的同时大概率使这 5 帧处于同一镜头下，方便数据集收集．收集完原始数据之后对数据进行筛选，若五元组中任意相邻 2 帧的 PSNR（Peak Signal-to-Noise Ratio）<15 或 SSIM<0.5，则认为 2 帧相似性过低，同理，若任意相邻 2 帧的 PSNR>35，认为 2 帧相似性过高．相似性过低说明 2 帧可能并非出自同一镜头，画面内容发生了较大变化，而相似性过高会导致数据之间的高相关性影响模型的泛化能力，因此，抛弃相似性过低或过高的五元组．之后还经过人工审核，过滤掉数值符合上述设定但实际上并非同一镜头的五元组，最终每部电影收集 100组五元组．图 6 展示了部分数据集图片，数据集中的主体除了现实生活中存在的物体、人物等，还包括了艺术作品中天马行空的各类形象，例如，各种拟人化动物、拥有特异外观的人物等．数据集中包含的场景十分广泛，有各种明亮、黑暗场景．相邻 2 张图片间隔 8 帧，也有较大位移．

为了更好进行训练，采样的数据采用了图 7 所示的帧匹配方式．具体而言，模型需要输入 1 帧上过色的图片及 1 帧待着色的线稿，输出 1 帧上完色的图片．所以，用线稿提取算法对五元组中的所有帧提取线稿，将彩色帧与线稿帧一一配对．对于每个五元组，共能得到 25（5×5）对输入帧，因此，800 组五元组共能配对产生 20 000 对输入帧二元组．因为帧与帧之间有所间隔，可能会出现参考帧中无而线稿帧中有的对象．针对这种现象，笔者考虑了二元组中线稿对应的标签帧为二元组中参考帧的配对模式，这样当参考帧中没有对象参考颜色时，模型也能够根据以往的学习经验，给出合理的着色结果，提高模型的泛化能力．

最终将五元组配对成了二元组，二元组中 2 帧的距离为 0（即线稿的标签帧与参考帧相同）的共有 4 000对，距离为 8 的共有 6 400 对，距离为 16 的共有 4 800 对，距离为 24 的共有 3 200 对，距离为 32 的共有 1 600 对，分布如图 8 所示．

图6　数据集示例

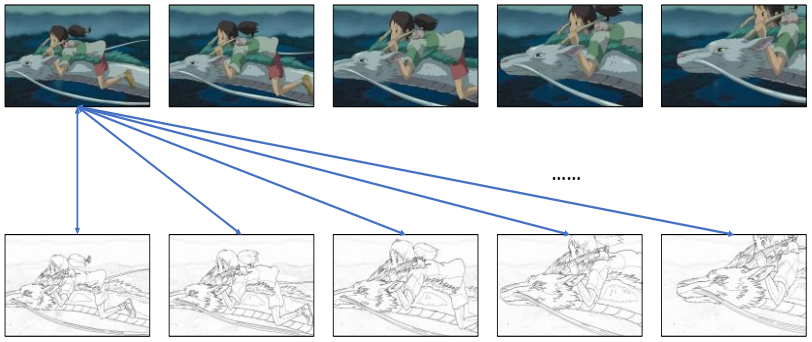



图7　数据配对

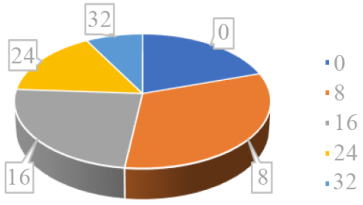



图8　帧距离统计

在训练过程中随机将20 000对数据进行划分，训练集与测试集的比例为8∶2，也就是16 000对作为训练集，剩下的4 000对作为测试集. 训练集对与测试集对无重叠.

### 4.2　学习策略

在网络训练时将epoch设置为600，使用Adam[38]作为优化器对网络进行优化，初始将学习率设置为$1\times10^{-4}$，经过300个epoch之后将学习率衰减为$1\times10^{-5}$. 为了符合Transformer结构的输入维度，输入网络图片的分辨率被调整为448×448. 网络由PyTorch深度学习框架构建，并在单个3 080 Ti上对网络进行训练和测试. 在训练阶段batch设置为4，测试阶段batch设置为1，大约150 h网络收敛.

### 4.3　评价指标

(1)PSNR

本文使用PSNR来评估生成图片的质量. PSNR的值越高，表示生成图像的质量越高. PSNR计算公式如下：

$$\mathrm{PSNR}(\boldsymbol{y},\hat{\boldsymbol{y}})=10\times\log_{10}\left(\frac{\mathrm{MAX}_I^2}{\mathrm{MSE}(\boldsymbol{y},\hat{\boldsymbol{y}})}\right) \tag{6}$$

其中，$\mathrm{MAX}_I^2$表示像素点的可能最大值；MSE(Mean Square Error)为真实图像与预测图像之间的均方误差，公式如下：

$$\mathrm{MSE}(\boldsymbol{y},\hat{\boldsymbol{y}})=\frac{1}{3mn}\sum_{\mathrm{R,G,B}}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left\|\boldsymbol{y}(i,j)-\hat{\boldsymbol{y}}(i,j)\right\|^2 \tag{7}$$

其中，$\boldsymbol{y}$表示预测图像；$\hat{\boldsymbol{y}}$表示真实图像；$m$，$n$表示图像大小.

(2)SSIM(Structure SIMilarity index measure)

本文使用SSIM来评估图片的结构相似性. SSIM分别从2张图像中提取亮度、对比度、结构3个关键特征，来度量2张图像的相似性. SSIM取值范围为[0, 1]，值越大，表示2张图之间的相似性越高. SSIM计算公式如下：

$$\mathrm{SSIM}(\boldsymbol{y},\hat{\boldsymbol{y}})=\frac{\left(2\mu_{\hat{y}}\mu_y+c_1\right)\left(2\sigma_{\hat{y}y}+c_2\right)}{\left(\mu_{\hat{y}}^2+\mu_y^2+c_1\right)\left(\sigma_{\hat{y}}^2+\sigma_y^2+c_2\right)} \tag{8}$$

$$c_1=\left(k_1L\right)^2 \tag{9}$$

$$c_2=\left(k_2L\right)^2 \tag{10}$$

其中，$\mu_y$，$\mu_{\hat{y}}$分别表示图像$\boldsymbol{y}$，$\hat{\boldsymbol{y}}$的平均值；$\sigma_y^2$，$\sigma_{\hat{y}}^2$分别表示图像$\boldsymbol{y}$，$\hat{\boldsymbol{y}}$的方差；$\sigma_{\hat{y}y}$表示图像$\hat{\boldsymbol{y}}$和图像$\boldsymbol{y}$的协方差；$c_1$，$c_2$为常数，避免分母接近于0时造成的计算不稳定；$k_1$取0.01，$k_2$取0.03；$L$为像素值的动态范围.

### 4.4 对比模型

本文算法与其他先进的着色算法进行比较，具体选取了Ref[25]、Stas[21]、Tcvc[31]和SGA[32]模型来和本文模型进行比较. 所有对比模型使用本文数据集重新训练并测试.

定量比较的结果如表1所示. Stas(Style transfer for anime sketches)算法的性能最低，可能是因为生成的辅助灰度图不够好影响了最终着色，Ref(Reference-based sketch image colorization)算法由于其自监督形式，使得难以适应变化较大的帧，所以指标不高，Tcvc(Temporally coherent video colorization)和SGA(Stop-Gradient Attention)算法由于其产生的模糊使得指标低于提出的算法.

**表1 不同算法的对比结果**

| 算法 | 输入类型 | PSNR | SSIM |
|---|---|---|---|
| Stas | Sketch+Reference | 10.04 | 0.350 |
| Ref | Sketch+Reference | 17.50 | 0.589 |
| Tcvc | Sketch+Reference | 23.16 | 0.796 |
| SGA | Sketch+Reference | 24.64 | 0.918 |
| 本文算法 | Sketch+Reference | 26.29 | 0.932 |

图9展示了各个算法的着色结果，其中，最后1行GT表示标签. Stas算法能够大致匹配物体的主要颜色，但是色彩变化较为剧烈，导致同一语义对象色彩发生较大变化，例如，最后1张人物的手臂出现较大的黑色色块. Ref作为自监督算法，对于$\boldsymbol{I}_{\mathrm{r}}$和$\boldsymbol{I}_{\mathrm{s}}$距离较大的帧无法得到较好结果，导致图片模糊. Tcvc算法的着色图片产生伪影，导致整张图片模糊，清晰度不够高. SGA在细节部分存在不精细的问题，所以指标略低于Cross-CNN. Cross-CNN将$\boldsymbol{I}_{\mathrm{r}}$和$\boldsymbol{I}_{\mathrm{s}}$同时进行卷积，保证了色域的一致性，交叉注意力的应用使颜色更加接近标签，着色结果存在噪声像素和伪影较少，在所有算法中有最好的视觉效果.

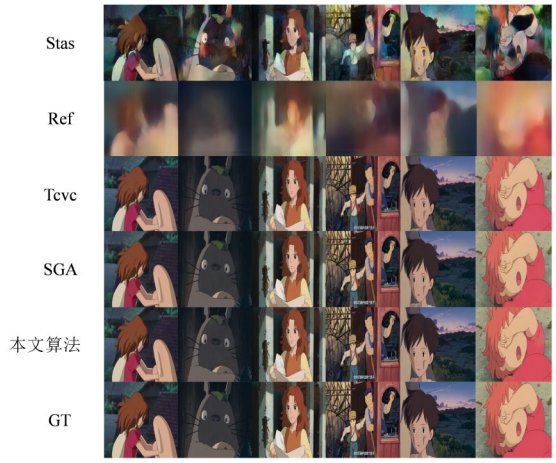



图9 不同算法的比较

在实际使用中，画师可能会随时修改对象某部分颜色，比如人物的头发、衣服颜色等信息. 为了更好贴近实际应用，替换测试数据集中人物的头发颜色、衣服颜色，来测试各模型对于颜色变化的敏感程度，即模型对参考帧的学习能力. 实验结果如表2所示，本文提出的模型取得了最好的性能.

**表2 参考帧学习能力测试**

| 算法 | 输入类型 | PSNR | SSIM |
|---|---|---|---|
| Stas | Sketch+Reference | 10.79 | 0.434 |
| Ref | Sketch+Reference | 15.22 | 0.558 |
| Tcvc | Sketch+Reference | 16.01 | 0.687 |
| SGA | Sketch+Reference | 18.72 | 0.687 |
| 本文算法 | Sketch+Reference | 22.69 | 0.911 |

图10可视化了不同算法对于参考帧色彩学习的测试结果，第一列的红色框表示修改颜色的区域. 可以发现，当修改参考帧中部分颜色时，Stas算法、Ref算法和SGA算法没有学习到参考帧中的颜色信息，导致部分区域着色结果未与参考帧保持一致，Tcvc算法的结果产生泛白，造成着色结果较差. 研究的算法能够按照参考帧的颜色信息进行相应更改，使对象颜色与参考帧中颜色同对象始终保持一致，且着色质量较高.

### 4.5 消融实验

本节探索了在线稿着色任务中使用交叉注意力的


图10 参考帧学习结果可视化

效果．具体而言，网络通过将$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$的来源设置为$\boldsymbol{I}_r$、$\boldsymbol{I}_s$或叠加图片$\boldsymbol{x}$，并与不带有交叉注意力的原始模型进行对比，来验证交叉注意力的作用．实验结果如表3所示，以第二行方法为例，该方法$\boldsymbol{I}_r$-$\boldsymbol{K}$表示$\boldsymbol{K}$矩阵的来源为$\boldsymbol{I}_r$，$\boldsymbol{x}$-$\boldsymbol{Q}$表示$\boldsymbol{Q}$矩阵的来源为$\boldsymbol{x}$，$\boldsymbol{x}$-$\boldsymbol{V}$表示$\boldsymbol{V}$矩阵的来源为$\boldsymbol{x}$，正如图5所示．其他方法以此类推．No-Cross表示未使用交叉注意力机制．可以发现，只要使用了交叉注意力，最终得到的指标都较原始模型更高，这证明了交叉注意力在线稿着色任务中的有效性．图11为有无交叉注意力机制的可视化结果．第一列为使用了交叉注意力的结果，第二列未使用交叉注意力，第三列为标签图片．红色框标注了差异较大的地方．可以看到，使用了交叉注意力机制能使着色结果更加准确．

**表3　不同交叉注意力设置的比较结果**

| 方法 | PSNR | SSIM |
|---|---|---|
| $\boldsymbol{I}_r$-$\boldsymbol{Q}$<br>$\boldsymbol{x}$-$\boldsymbol{K}$<br>$\boldsymbol{x}$-$\boldsymbol{V}$ | 26.18 | 0.932 |
| $\boldsymbol{I}_r$-$\boldsymbol{K}$<br>$\boldsymbol{x}$-$\boldsymbol{Q}$<br>$\boldsymbol{x}$-$\boldsymbol{V}$ | 26.29 | 0.932 |
| $\boldsymbol{I}_s$-$\boldsymbol{Q}$<br>$\boldsymbol{x}$-$\boldsymbol{K}$<br>$\boldsymbol{x}$-$\boldsymbol{V}$ | 26.16 | 0.932 |
| $\boldsymbol{I}_s$-$\boldsymbol{K}$<br>$\boldsymbol{x}$-$\boldsymbol{Q}$<br>$\boldsymbol{x}$-$\boldsymbol{V}$ | 26.25 | 0.932 |
| No-Cross | 25.28 | 0.927 |


图11　有无交叉注意力机制的差异可视化

此外，无论产生$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$矩阵的来源帧如何变换，使用了交叉注意力的模型的SSIM指标都相同．然而，如果$\boldsymbol{K}$矩阵由$\boldsymbol{I}_r$产生，则能取得最高的PSNR指标．这是因为两帧之间纹理信息的变化引起色彩的变化，相较于只包含纹理信息的$\boldsymbol{I}_s$，参考帧中既包含了丰富的颜色信息，也包含了纹理信息，因此将$\boldsymbol{I}_r$设置为$\boldsymbol{K}$可以提供更多的有用信息，对于变化较大的两帧能起到很好的联系作用．

### 4.6　跨帧着色

在序列着色过程中，$\boldsymbol{I}_r$与$\boldsymbol{I}_s$的距离会影响模型的着色效果，$\boldsymbol{I}_r$和$\boldsymbol{I}_s$距离越近，意味着画面变化较小，距离越大画面变化越大，难度也越大．为了探究这种影响，在测试中对$\boldsymbol{I}_r$和$\boldsymbol{I}_s$的间隔进行了不同的设置，并将本文的算法性能与其他同样使用参考帧作为输入的算法进行了比较．其中，对比算法排除了Ref算法，因为其自监督算法自定义远距离参考帧会导致效果很差．实验结果如表4所示，不难发现，随着$\boldsymbol{I}_r$与$\boldsymbol{I}_s$的距离间隔逐渐变大，除了SGA算法，其他算法的性能指标都呈下降趋势．这是由于当$\boldsymbol{I}_r$和$\boldsymbol{I}_s$之间的距离较远时，画面的变化较大，难以捕捉到它们之间的联系．

**表4　不同距离帧的对比结果**

| 算法 | 距离 | PSNR | SSIM |
|---|---|---|---|
| Stas | 8 | 10.09 | 0.352 |
| Tcvc | 8 | 21.23 | 0.786 |
| SGA | 8 | 23.95 | 0.906 |
| Copy | 8 | 14.07 | 0.730 |
| 本文算法 | 8 | 25.07 | 0.930 |
| Stas | 16 | 10.01 | 0.348 |
| Tcvc | 16 | 19.72 | 0.741 |
| SGA | 16 | 23.96 | 0.906 |
| Copy | 16 | 12.77 | 0.679 |
| 本文算法 | 16 | 24.17 | 0.916 |
| Stas | 24 | 9.88 | 0.345 |
| Tcvc | 24 | 19.04 | 0.716 |
| SGA | 24 | 23.92 | 0.905 |
| Copy | 24 | 12.18 | 0.650 |
| 本文算法 | 24 | 23.71 | 0.907 |
| Stas | 32 | 10.09 | 0.350 |
| Tcvc | 32 | 18.31 | 0.696 |
| SGA | 32 | 23.96 | 0.906 |
| Copy | 32 | 11.53 | 0.635 |
| 本文算法 | 32 | 23.21 | 0.900 |

本文提出的算法性能指标相对较高且下降趋势相对平缓，在不同帧距离对比中均取得较好效果．虽然SGA算法性能变化始终平稳，但根据分析，其对于参考帧的学习能力较差．为进一步验证算法的有效性，实验还将$\boldsymbol{I}_r$直接与标签进行比较，以模拟将$\boldsymbol{I}_r$不做任何处理复制到线稿上的效果，并将该方法记为Copy．可以发现，与Copy相比，Cross-CNN在不同距离下取得更好性能，进一步证明了算法的有效性和适用性．

为了展示Cross-CNN模型跨帧着色的性能，在图12中展示了$\boldsymbol{I}_r$以及$\boldsymbol{I}_s$．具体来说，图中第1行和第2行分别代表$\boldsymbol{I}_r$和$\boldsymbol{I}_s$，即网络的输入，第3行为网络的输出，第4

行为标签. Cross-CNN 展现出了非常好的性能，即使在 $\boldsymbol{I}_{\mathrm{r}}$ 和 $\boldsymbol{I}_{\mathrm{s}}$ 之间存在较大的变化时，网络能够准确完成着色任务. 例如，在图中第 1 列图片中，参考帧中包含 2根手指，而线稿中只包含 1 根手指. 通过使用交叉注意力机制，Cross-CNN 能准确识别对象并进行着色. 在数据集中还包含了距离为 0 的训练数据，这使得 Cross-CNN 可以很好处理那些在参考帧中未出现而在线稿中出现的对象，如图中第 4 列所示. 在这个例子中，参考帧中只有 1 个人物，但线稿中却有 2 个人物，网络也能准确完成着色任务，证明了网络的泛化能力非常强.


图 12 跨帧着色结果

### 4.7 创意编辑

本文算法的目标是训练可实现动画线稿着色的网络，在实际应用时用户可输入手绘的场景线稿，也可根据自身需求对已有场景线稿作为底稿进行修改、编辑，绘制符合需求的线稿作为网络输入并生成着色场景，增强动画着色应用的灵活性.

本节实验从测试集中选取了 2 张线稿，分别采用擦除、添加等方式对原始线稿进行编辑. 图 13 展示了将编辑后的线稿作为网络输入的着色效果，第 1 列为参考帧，第 2 列为原始线稿，第 3 列为经过修改之后的线稿，红色框代表编辑区域，其中，对第 1 行的线稿天空部分擦除了部分云，对第 2 行的线稿增加了涟漪线稿，第 4 列为网络输出. 着色效果表明，本文提出的算法能适用于多种方式编辑，在实际应用中，用户可通过编辑线稿实现场景结构的构思与创作.

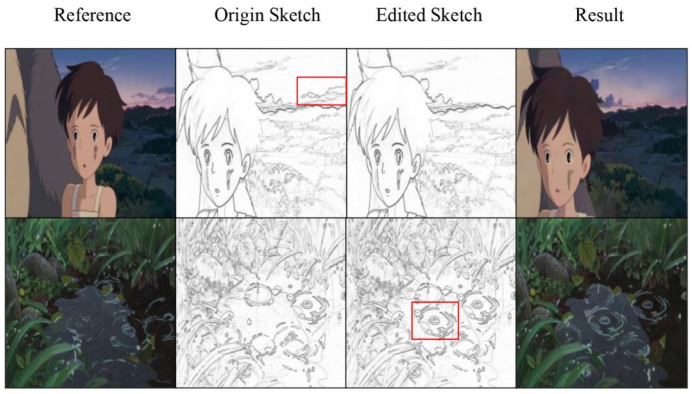



图 13 创意编辑

## 5 结语

本文使用了基于 CNN 和 Transformer 的混合模型 Cross-CNN 来给线稿着色，卷积能够更好地提取局部信息，Transformer 提取全局特征的能力更强. 针对于两帧输入，设计了交叉注意力，使得间隔较远的帧也能很好匹配特征. 实验在自制的数据集上进行，结果证明了本文方法的有效性，Cross-CNN 的 PSNR 值达到了 26.29，SSIM 值达到了 0.932，在所有算法中取得了最好的视觉效果.

### 参考文献

[1] ZENG R. Research on the application of computer digital animation technology in film and television[J]. Journal of Physics: Conference Series, 2021, 1915(3): 032047.

[2] ZHANG Q, WANG B, WEN W, et al. Line art correlation matching feature transfer network for automatic animation colorization[C]//2021 IEEE Winter Conference on Applications of Computer Vision (WACV). Piscataway: IEEE, 2021: 3871-3880.

[3] SHI M, ZHANG J Q, CHEN S Y, et al. Reference-based deep line art video colorization[J]. IEEE Transactions on Visualization and Computer Graphics, 2023, 29(6): 2965-2979.

[4] YOO S, BAHNG H, CHUNG S, et al. Coloring with limited data: Few-shot colorization via memory augmented networks[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2019: 11275-11284.

[5] CASEY E, PÉREZ V, LI Z R. The animation transformer: Visual correspondence via segment matching[C]//2021 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE, 2021: 11303-11312.

[6] LI S Y, ZHAO S Y, YU W J, et al. Deep animation video interpolation in the wild[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2021: 6583-6591.

[7] HORIUCHI T, HIRANO S. Colorization algorithm for grayscale image by propagating seed pixels[C]//Proceedings 2003 International Conference on Image Processing. Piscataway: IEEE, 2003: 1-457.

[8] LEVIN A, LISCHINSKI D, WEISS Y. Colorization using optimization[J]. ACM Transactions on Graphics, 23(3): 689-694.

[9] QU Y G, WONG T T, HENG P A. Manga colorization[J]. ACM Transactions on Graphics, 25(3): 1214-1220.

[10] SÝKORA D, DINGLIANA J, COLLINS S. LazyBrush: Flexible painting tool for hand-drawn cartoons[J]. Computer Graphics Forum, 2009, 28(2): 599-608.

[11] YATZIV L, SAPIRO G. Fast image and video colorization using chrominance blending[J]. IEEE Transactions on Image Processing, 2006, 15(5): 1120-1129.

[12] GOODFELLOW I J, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets[C]//Proceedings of the 27th International Conference on Neural Information Processing Systems. New York: ACM, 2014: 2672-2680.

[13] ISOLA P, ZHU J Y, ZHOU T H, et al. Image-to-image translation with conditional adversarial networks[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2017: 5967-5976.

[14] MIRZA M, OSINDERO S. Conditional generative adversarial nets[EB/OL]. (2014-11-06)[2023-07-01]. http://arxiv.org/abs/1411.1784.

[15] ZHU J Y, PARK T, ISOLA P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[C]//2017 IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE, 2017: 2242-2251.

[16] NAZERI K, NG E, EBRAHIMI M. Image colorization using generative adversarial networks[M]//Articulated Motion and Deformable Objects. Cham: Springer International Publishing, 2018: 85-94.

[17] SU J W, CHU H K, HUANG J B. Instance-aware image colorization[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 7965-7974.

[18] 金正猛, 周晨. 基于耦合全变差的快速图像着色算法[J]. 电子学报, 2016, 44(10): 2364-2369.
JIN Z M, ZHOU C. A fast coupled total variation algorithm for image colorization[J]. Acta Electronica Sinica, 2016, 44(10): 2364-2369. (in Chinese)

[19] 李洪安, 郑峭雪, 张婧, 等. 结合Pix2Pix生成对抗网络的灰度图像着色方法[J]. 计算机辅助设计与图形学学报, 2021, 33(6): 929-938.
LI H A, ZHENG Q X, ZHANG J, et al. Pix2Pix-based grayscale image coloring method[J]. Journal of Computer-Aided Design & Computer Graphics, 2021, 33(6): 929-938. (in Chinese)

[20] RONNEBERGER O, FISCHER P, BROX T. U-Net: Convolutional networks for biomedical image segmentation[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015: 234-241.

[21] ZHANG L M, JI Y, LIN X, et al. Style transfer for anime sketches with enhanced residual U-net and auxiliary classifier GAN[C]//2017 4th IAPR Asian Conference on Pattern Recognition (ACPR). Piscataway: IEEE, 2017: 506-511.

[22] ZHANG L M, LI C Z, WONG T T, et al. Two-stage sketch colorization[J]. ACM Transactions on Graphics, 2018, 37(6): 1-14.

[23] ZHANG B, HE M M, LIAO J, et al. Deep exemplar-based video colorization[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2019: 8044-8053.

[24] LIU X T, WU W L, LI C Z, et al. Reference-guided structure-aware deep sketch colorization for cartoons[J]. Computational Visual Media, 2022, 8(1): 135-148.

[25] LEE J, KIM E, LEE Y, et al. Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 5800-5809.
[26] ZHANG Q, WANG B, WEN W, et al. Line art correlation matching feature transfer network for automatic animation colorization[C]//2021 IEEE Winter Conference on Applications of Computer Vision (WACV). Piscataway: IEEE, 2021: 3871-3880.
[27] ZHANG J S, ZHU S Q, LIU K X, et al. UGSC-GAN: User-guided sketch colorization with deep convolution generative adversarial networks[J]. Computer Animation and Virtual Worlds, 2022, 33(1): e2032.
[28] CHO Y, LEE J, YANG S, et al. Guiding users to where to give color hints for efficient interactive sketch colorization via unsupervised region prioritization[C]//2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). Piscataway: IEEE, 2023: 1818-1827.
[29] LI X Y, ZHANG B, LIAO J, et al. Deep sketch-guided cartoon video inbetweening[J]. IEEE Transactions on Visualization and Computer Graphics, 2022, 28(8): 2938-2952.
[30] HENSMAN P, AIZAWA K. cGAN-based manga colorization using a single training image[C]//2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). Piscataway: IEEE, 2017: 72-77.
[31] THASARATHAN H, NAZERI K, EBRAHIMI M. Automatic temporally coherent video colorization[C]//2019 16th Conference on Computer and Robot Vision (CRV). Piscataway: IEEE, 2019: 189-194.
[32] LI Z K, GENG Z Y, KANG Z, et al. Eliminating gradient conflict in reference-based line-art colorization[C]//European Conference on Computer Vision. Cham: Springer, 2022: 579-596.
[33] LIN X X, WANG X, LI F, et al. Example-based image recoloring in an indoor environment[J]. Computer Animation and Virtual Worlds, 2019, 31(2): e1917.
[34] LIU S F, ZHONG G Y, DE MELLO S, et al. Switchable temporal propagation network[C]//European Conference on Computer Vision. Cham: Springer, 2018: 89-104.
[35] VONDRICK C, SHRIVASTAVA A, FATHI A, et al. Tracking emerges by colorizing videos[C]//Computer Vision-ECCV 2018: 15th European Conference. New York: ACM, 2018: 402-419.
[36] CHENG S N, CHEN Y J, CHIU W C, et al. Adaptively-realistic image generation from stroke and sketch with diffusion model[C]//2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). Piscataway: IEEE, 2023: 4043-4051.
[37] CHEN J N, LU Y Y, YU Q H, et al. TransUNet: Transformers make strong encoders for medical image segmentation[EB/OL]. (2024-02-08)[2023-07-01]. http://arxiv.org/abs/2102.04306.
[38] KINGMA D P, BA J. Adam: A method for stochastic optimization[EB/OL]. (2014-12-22)[2023-07-01]. http://arxiv.org/abs/1412.6980.

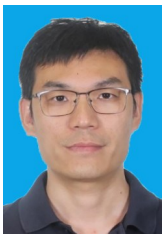
## 作者简介

**余毅丰** 男，1998 年 8 月出生，浙江省宁波人. 宁波大学信息科学与工程学院硕士研究生. 主要研究方向是计算机视觉、图像着色.
E-mail: 2011082343@nbu.edu.cn

**钱江波** 男，1974 年 7 月出生，浙江省宁波人. 宁波大学信息科学与工程学院教授、博士生导师. 主要研究方向为计算机视觉、数据挖掘.
E-mail: qianjiangbo@nbu.edu.cn

**严迪群** 男，1979 年 7 月出生，浙江省宁波人. 现为宁波大学信息科学与工程学院副教授. 主要研究方向为深度学习、计算机视觉.
E-mail: yandiqun@nbu.edu.cn

**王 翀** 男，1985 年 2 月出生，浙江省宁波人. 现为宁波大学信息科学与工程学院副教授. 主要研究方向为计算机视觉、图像/视频处理.
E-mail: wangchong@nbu.edu.cn



**董 理** 男，1990 年 8 月出生，河南省周口人. 宁波大学信息科学与工程学院副研究员. 主要研究方向为多媒体内容. 中国电子学会会员编号：E190036628M.
E-mail: dongli@nbu.edu.cn