Ⅰ·视频擦除简介

视频擦除（Video Inpainting），即在一段连续视频中涂抹一个区域或一个运动物体，并用生成的方式将其掩盖填补回来，要求尽量的不留痕迹（即肉眼难以察觉）。

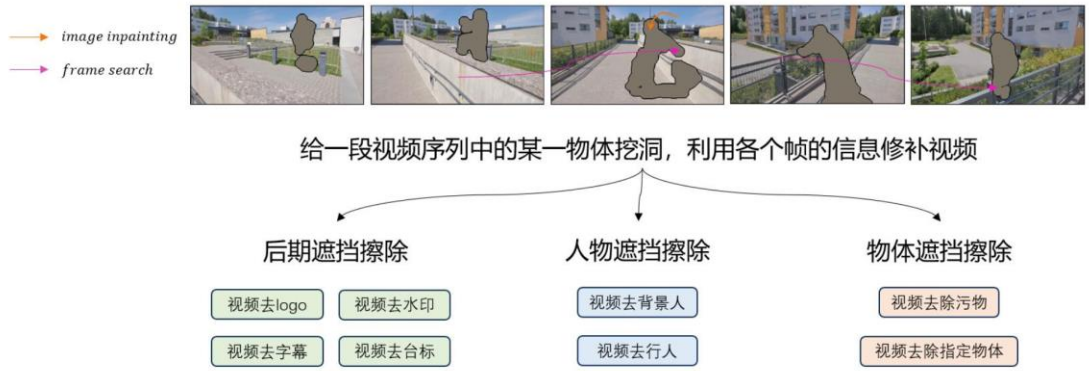


图 1-1 Video Inpainting 介绍

如上图所示的例子，在一段视频序列中有一个运动的人物，我们用 mask(遮罩)的方式将人物给涂抹掉后，要基于视频的其他内容信息将涂抹部分填充回来，且最终修复完好的视频应该具有人物“消失掉”的效果。这是一个很有意思的任务，可衍生出的落地玩法是非常多的。譬如对于视频后期遮挡的擦除，可实现去 logo、去水印、去字幕、去台标等任务；再比如对视频人物遮挡的擦除，可实现去背景人、去行人等功能；它也可以

被用来做指定视频物体的擦除，例如去污物或者消除特定物体等等。

不过，Video Inpainting 具有一定的技术难点，主要体现在它既需要在自身帧上作空域信息的搜寻（即 Image Inpainting 部分），也需要在相邻帧上作时域信息的搜寻（即 Memory Search+Read 部分），并最终将两部分信息整合并填补回原始帧中。

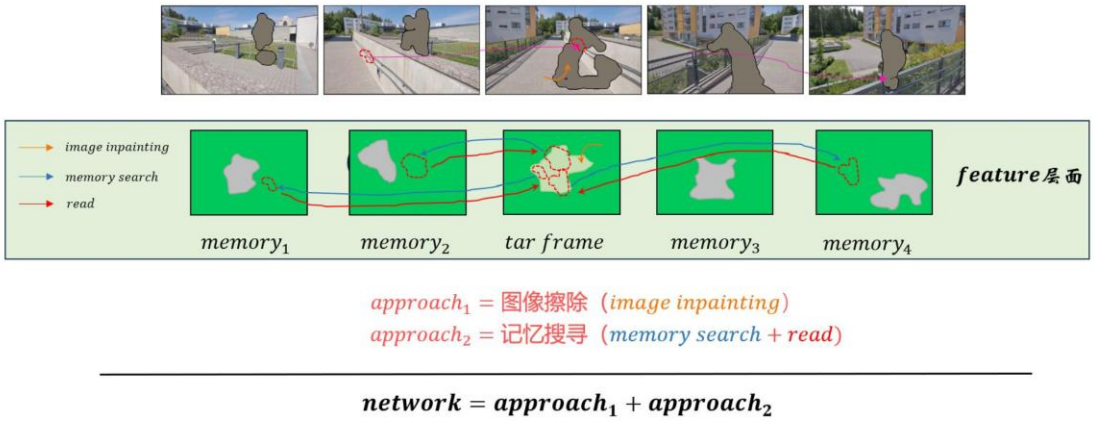


图 1-2 Video Inpainting 难点

具体来说(如上图所示)，以第三帧人物轮廓的填补进行分析，最佳修复方式应该是：其腿部部分可以沿用周围马路上的像素点 copy 过来，因为很明显马路上的像素块是差不多的；而其上部分栏杆和房屋位置的信息应该是采自第二帧中匹配的像素点 copy 过来，因为

这些偏细节的内容若能在其他帧中搜寻到且复制回来会更为准确。

因此，如果往更一般的情况推广，Video Inpainting 首先应考虑当前帧缺失的信息是否有在其它帧中“露出”，如果有匹配上的信息就将其填补回来（这有利于准确的细节内容重构）；剩下的缺失信息再通过 Image Inpainting 的方式填补（有利于填补掉未修复完全的内容）。

综上可以看出，Video Inpainting 需要整合多维度获取的信息，并有效的转化和填补回修复图像上，这是比较复杂且困难的部分。这些技术难点也是目前 Video Inpainting 发展缓慢、落地应用比较滞后的原因。

另外一方面，我们需要一些学术界的方法作为技术突破的指引。因此确认这一任务方向后，我去做了很多相关调研。从调研结果来看，Video Inpainting 的相关论文集中在 19-20 年间，数目非常少。从思想上来说，Video Inpainting 主要分为两种擦除方式，一种是显式擦除，也就是直接作用在图像上，从像素层面进行修复；另一种是隐式擦除，也就是作用在图像编码上，从特征层面进行修复。

Source: 研究笔记（seeprettyface.com）

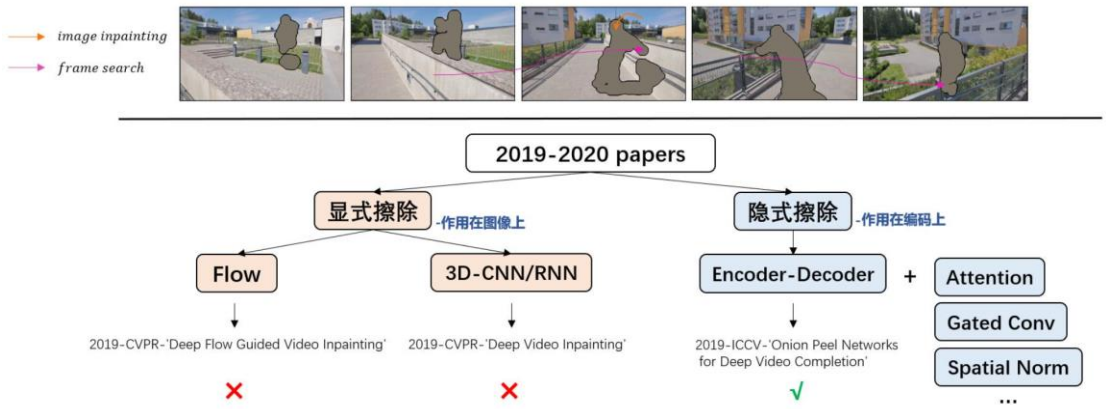


图 1-3 Video Inpainting 方法

先简单介绍下显式擦除，也就是直接构建“图到图“的网络设计模式。一种是基于光流（Flow）计算的做法，也就是基于前后帧的差异计算出像素的“运动趋势”，利用这种“趋势”预测出颜色的传播从而填补掉缺失的 mask 块；另一种是基于 3D-CNN/RNN 的设计，它直接将时序上的帧按通道数堆积在一起，形成一个大型矩阵进行卷积计算，这种做法需要进行一定的改良才能避免高维卷积带来的潜在隐患（譬如计算浪费高、数据依赖严重等等），其中一种有效的改良方式是利用多头注意力机制分 block 进行计算等。

另外一种设计模式是隐式擦除，其首先利用“Encoder-Decoder”架构获取了图像的深度表征，然后通过对图像表征的系列修补操作（譬如 attention，

gated-convolution，region-normalization 等等）后，再映射(decoder)回图像，生成出修补完成的图片帧。

II· 一种精巧好用的模型——OPN

通过多轮的调研和测试，我们发现基于显式擦除的网络较为笨重，占用内存高、计算开销较大，且遇到一些交叉遮挡、频繁切换的视频场景效果就会变糟，因此可改良的空间很小；而基于隐式擦除的网络小巧且精妙，效果相对稳健，具备一定的落地潜力。其中最具代表性的论文是 OPN（2019-ICCV-Onion Peel Networks for Deep Video Completion），它在测试数据上的表现是稳定且出色的，对于小块的擦除有着较好的隐身效果。

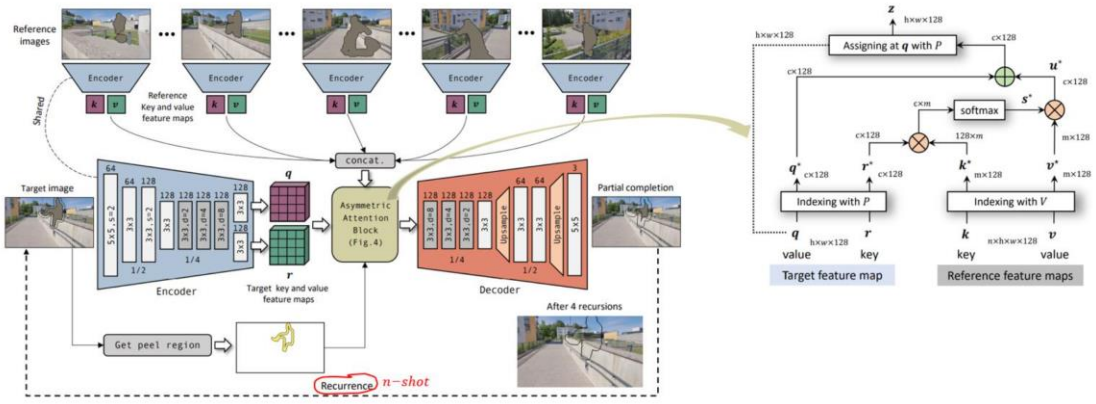


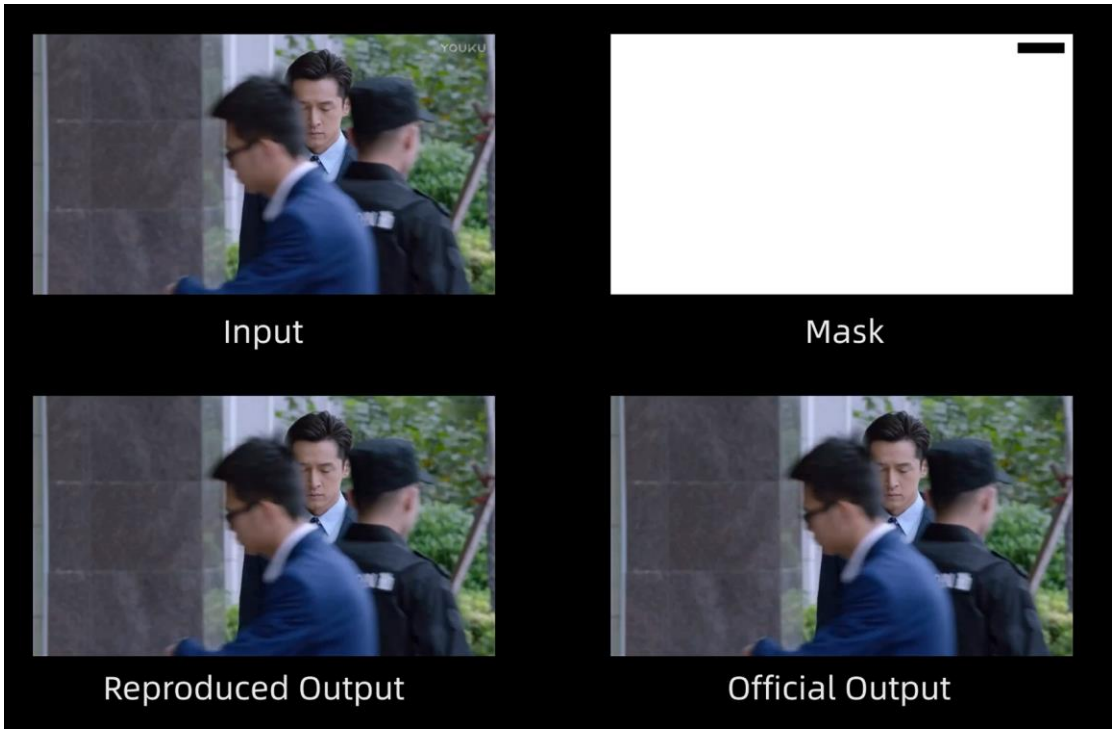
图 2-1 OPN 模型架构

Source: 研究笔记（seeprettyface.com）

于是，接下来需要实现的重点任务，就是去攻克OPN的训练源码复现，争取获得自己的视频擦除基线模型。先简单梳理一下OPN的模型架构逻辑，其左右两端分别是Encoder和Decoder模块，通过Encoder获得的图像表征有两部分，分别是q（表征注意力信息）和r（表征内容信息），将q和每一个采样的记忆帧的k（即记忆帧的注意力信息，通过同样的Encoder获得）匹配后，所获得的attention权重乘上记忆帧的v（记忆帧的表征内容信息）得到从该记忆帧获取的信息增补量u，然后将u加回到当前帧的r上就实现了一次特征层面的修补，最终将修补完成的新的表征通过Decoder输出便得到修复后的图像。另外，OPN还使用了一个小trick：由于其Encoder抓捕q和r表征的能力不强（这是后续的结论），所以其擦除采用了按圈擦除（记作n-shot，表示擦多次），譬如作者采用一次只修补8-pixel的像素宽度，一圈一圈向里填补。

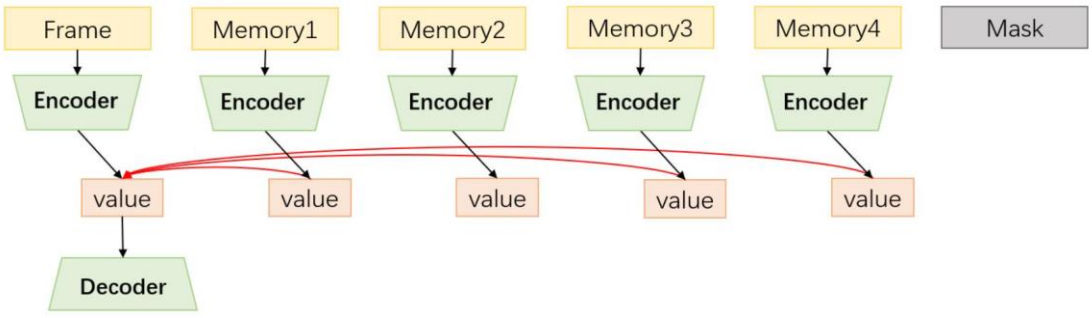
总而言之，尽管OPN在空域擦除上没有深化改良措施，但其核心贡献是它解决了视频擦除的第二个难题，也就是在时域上的记忆搜寻过程。这是具有开创性和重要突破意义的，因为很多做video inpainting的人都在想如何能在时域上搜刮信息，而OPN率先实现了

优秀的效果，因此其时域搜寻部分的训练模式是具备拓展价值的，复现其源码也就有必要性。



上述是复现的 OPN 模型与官方模型的擦除对比。可以看出 OPN 在小块的擦除上效果还是不错的，比如对于台标这种小块区域的擦除可以考虑采用 OPN 作为基线模型。

III· 对于 OPN 模型的一种改良思路



Source: 研究笔记 (seeprettyface.com)

图 3-1 OPN 模型缺陷

前面有提到过，OPN 的 Encoder 抓捕 q 和 r 表征的能力不强，得出这一结论的原因是当我们将上图中红色部分抹除掉（即删除掉时域搜索部分）后，OPN 的擦除能力大打折扣了，甚至落后于很多新型的 Image Inpainting 的模型效果。因此，改进模型的出发点是替换 OPN 的 Encoder 和 Decoder 架构，使其分别具有更强的信息提取能力和精细内容重建能力。通过阅读论文，我们引入了新型归一化组件 Region Normalization，因为其具有更强的捕捉图像结构信息的能力，同时结合 OPN 的时域推理框架，我们提出了具备更好性能的改良模型 RNoT。

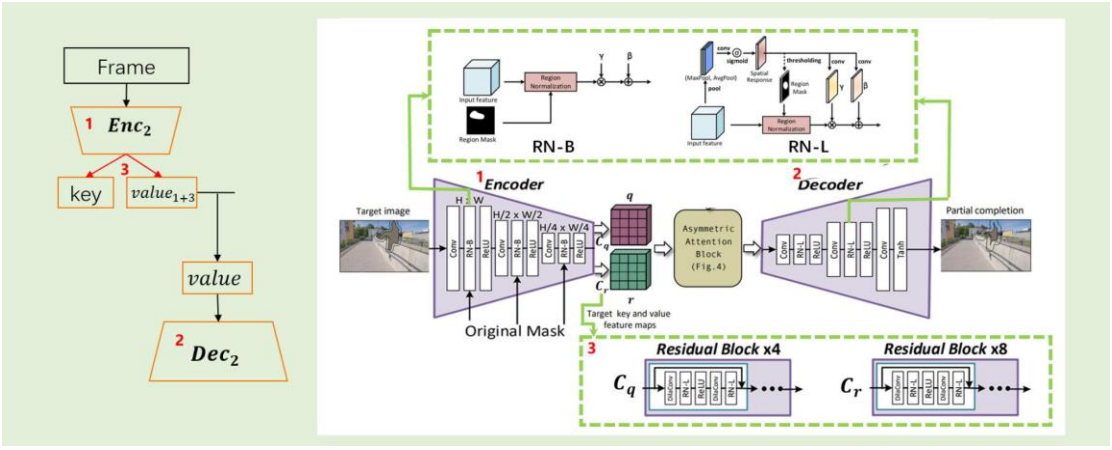


图 3-2 RNoT 架构示意图

上图是提出的 RNoT 的架构示意图，由于改进重点是在空域部分上，所以时域部分的流程图就没有再

画出（与 OPN 一致）。改进的部分主要是三点，第一个是 Encoder 里面加入了 RN-B 组件，第二个是 Decoder 里面加入了 RN-L 组件，其次提取 q 和 r 的部分分别采用了 4 层 Res-Block 和 8 层 Res-Block，以帮助其在时域推导的时候具有更好的变化弹性。

图 3-3 RNoT 与 OPN 对比一

这是一个 RNoT 与 OPN 在测试数据上的对比效果。左边第一行 Input 栏展示带有 Mask（灰色像素即为需要填补的区域）的输入帧，右边 Memory 里边展示了 4 帧可见的相邻帧(时域信息就在这些帧里面找)，左边第二行展示 OPN（按圈擦除）的输出效果，左边第三行展示 RNoT（单次擦除）的输出效果，左边第四行是参考的真实图像。

从结果可以看出，对于仅限于单次擦除的 RNoT，其还原效果也要优于按圈擦除的 OPN，主要体现在对于人物嘴巴、眼睛以及领带等细节内容的捕捉和重建上，RNoT 具备有更好的精度和准确度。

指标	OPN_{1-shot}	OPN_{n-shot}	RNoT
PSNR	40.2	41.7	41.4
FPS	16	2.28	16

图 3-4 RNoT 与 OPN 对比二

这是第二个在验证集上更详尽的对比。由于验证集的场景非常复杂，对于精细内容的捕捉和重建就更有难度。可以看出，如果都放在单次擦除的限制下比较，OPN 一次比较所能提取、重建回的信息依然有很多缺失（左边第三行），但是 RNoT 可以基本准确的还原回图像的主要信息（左边第二行）；如果放在不限次数的擦除下比较，OPN 的整体效果还是略胜 RNoT 一筹（左边第四行），但是其时间计算开销是远远高过 RNoT 的。也许有人会好奇，如果将 RNoT 也改成按圈

多次擦除会怎样。很遗憾，因为 RNoT 的计算方式是 Region Normalization(区域归一化)，即将图像当作整体去作归一化以提取结构信息，所以如果按圈擦除的话会破坏结构信息的统一性，导致圈与圈之间出现错位与扭曲。

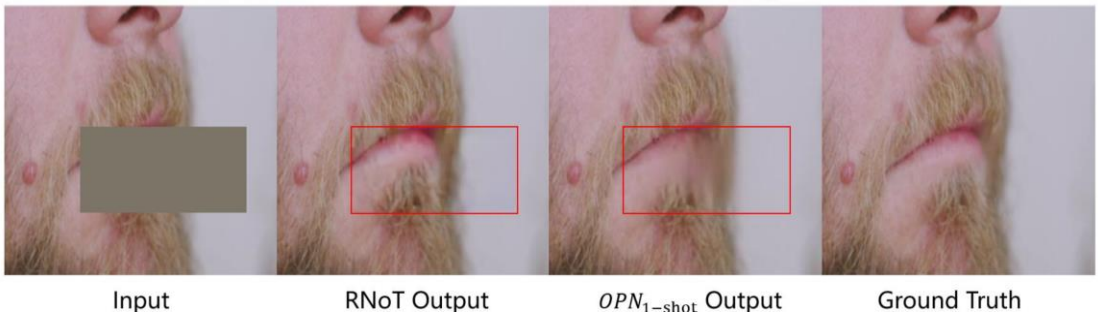


图 3-5 RNoT 细节展示

上图是一张更清晰的图片展示 RNoT 具备一定的高信息精度提取与重建能力。

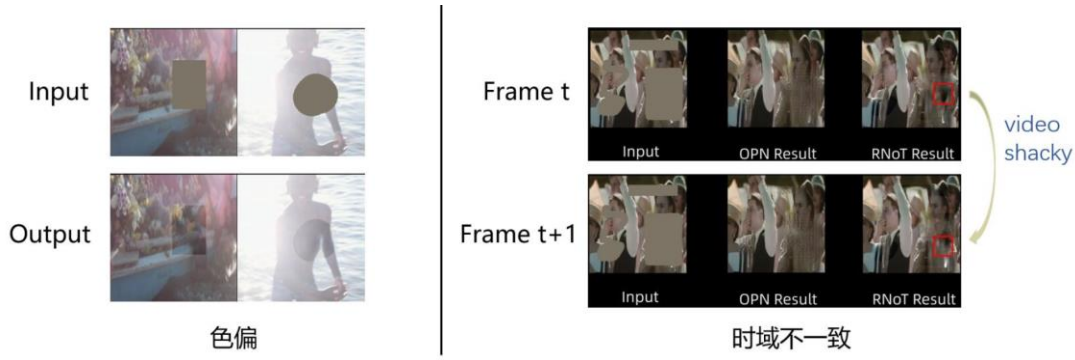


图 3-6 RNoT 的缺陷

当然 RNoT 也不是万能的，它也具备一定缺陷和进一步改良的空间。

首先第一个缺陷，也是 Region Normalization 本身就具有的缺陷，即容易出现色偏。特别是在浅色系背景或者强光条件下，RNoT 的还原内容会明显与周围内容出现不一致的颜色偏差，这在视频中会明显展露出来。究其原因可能是 RN 过分将注意力关注了在图像的结构信息上，而导致对颜色的不敏感。

第二个缺陷，也是 OPN 乃至整个隐式擦除架构容易具有的缺陷，就是时域不一致（前后帧由于局部像素变化大导致播放时产生异常抖动的效果），这在视频测试时时常出现。究其原因，是因为隐式的擦除架构都是单帧与单帧的独立比较，而忽略了帧的次序与连贯性；相比之下，显式的擦除模型（例如光流或 3D 卷积）就非常在意帧的时间顺序，因此其修复视频在时域上就会更为流畅与统一。总而言之，显式和隐式的擦除模型还是存在很多可以相互借鉴、共同改良的地方。

Ⅳ· 探索更实际好用的视频擦除框架

简单来说，以字幕擦除为例，由于其更注重时域一致性（即隐身效果）而不太重视内容重建，因此可以利用 3D-CNN+Transformer 的思想去做（类似于 STTN）。有趣的一点是，由于 3D-CNN 具有良好的时域感知性

（即字幕会在很多帧中重复出现所以 3D-CNN 较容易感知到字幕的位置），所以我们不需要引入 Mask、直接让模型自行感知字幕位置并进行擦除即可。

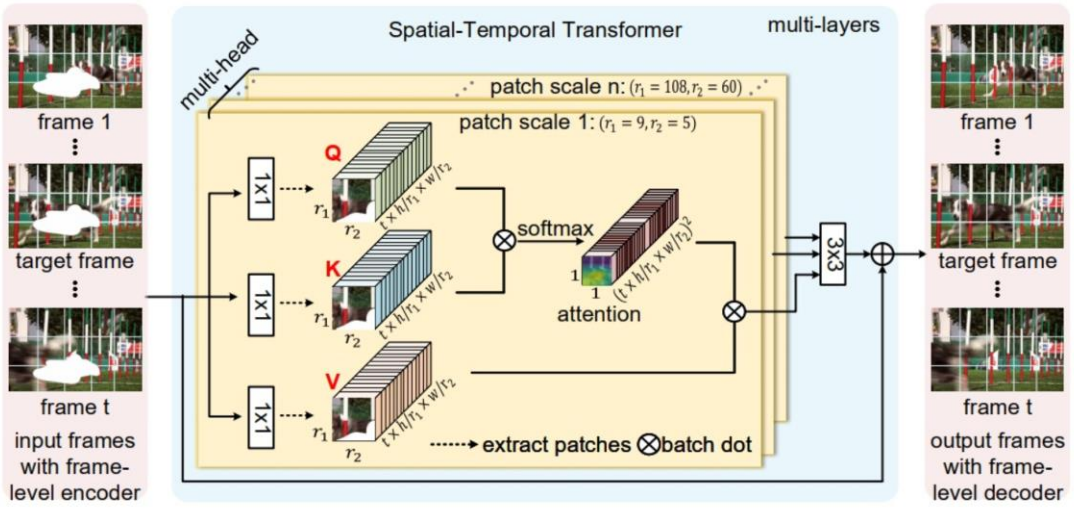


图 4-1 STTN 模型架构

当然仅仅使用 3D-CNN 还不足以让模型达到对于特定目标具有准确感知的能力，下面介绍一种非常简单的小技巧。

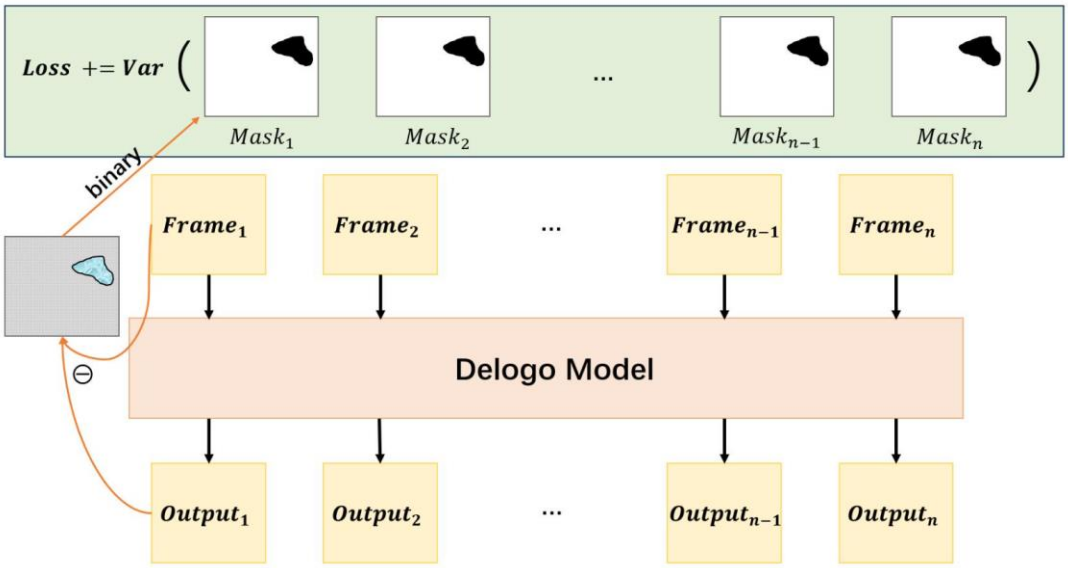


图 4-2 Logo 擦除：依据数据逻辑进行 Loss 的改良设计

以 Logo 擦除为例，由于 Logo 具有位置固定、内容固定的特点，因此我们可以对于每一组序列图片，计算出其输入和输出的差值二值化图像，并在时域上计算所有二值化图像的方差，最后将方差加在 Loss 上就可以了。理由是，如果模型擦除的位置或内容不一致的话，方差会比较大，最终拖累 loss 的值。实验表明，这种先验知识的引入对于提升效果还是挺有帮助的；对于其他类型的任务，我们也可以依据对应的数据逻辑进行相应的改良设计。

V. 模型训练

（1）训练数据

- 背景数据制作

搜集的至少 3000 片段的高清电影、TikTok 短片制作了数据集；

- 前景数据制作

1.字幕擦除：利用 ImageDraw 库生成随机样式、字体的文字，并模拟其变换；

2.图标擦除：利用 ImageDraw 库生成随机的像素区块，并模拟时域一致性（固定在视频中的某一个区域）；

3.动态图标擦除：利用 PR 软件制作闪烁、跳跃等字幕的动态特效，模拟动态图标的场景。

（2）训练过程

第 1 步：针对特定任务的时域感知训练，即让模型能感知到需被擦除的前景数据；

第 2 步：融合进擦除模型，进行端到端的微调训练。