10.10干的事：

1. 李沐的视频 ViT论文模型
2. 师兄推荐的论文
3. wct的代码

1 vit：直接使用标准nlp的transformer都可以实现cv操作（直接爆火，继alexnet后）

第一个论文：Intriguing Properties of Vision Transformers

第二个论文： An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

> ViT是2020年Google团队提出的将Transformer应用在图像分类的模型，虽然不是第一篇将transformer应用在视觉任务的论文，但是因为其模型“简单”且效果好，可扩展性强（scalable，模型越大效果越好），成为了transformer在CV领域应用的里程碑著作，也引爆了后续相关研究
>
> 把最重要的说在最前面，ViT原论文中最核心的结论是，当拥有足够多的数据进行预训练的时候，ViT的表现就会超过CNN，突破transformer缺少归纳偏置的限制，可以在下游任务中获得较好的迁移效果
>
> 但是当训练数据集不够大的时候，ViT的表现通常比同等大小的ResNets要差一些，因为Transformer和CNN相比缺少归纳偏置（inductive bias），即一种先验知识，提前做好的假设。CNN具有两种归纳偏置，一种是局部性（locality/two-dimensional neighborhood structure），即图片上相邻的区域具有相似的特征；一种是平移不变形（translation equivariance），$f(g(x)) = g(f(x))$，其中g代表卷积操作，f代表平移操作。当CNN具有以上两种归纳偏置，就有了很多先验信息，需要相对少的数据就可以学习一个比较好的模型

tip：模型简单：2500天tpuv3训练时间 草

把transformer应用到视觉问题的难处：硬件支持序列长度最大512； 我们要怎么把2d的图片弄成一个1d 的序列；（224*224个像素点，太大了不可行）

于是有一些人进行了一定的革新（先cnn，把输出的特征拉平就行） 或者 在输入图像上加一个window，控制输入图片的大小，；或者在图片的H和W两轴方向分别做一个selfattention等等； 但没有办法在现在的硬件配置上进行加速，因此没有很好的改善什么；

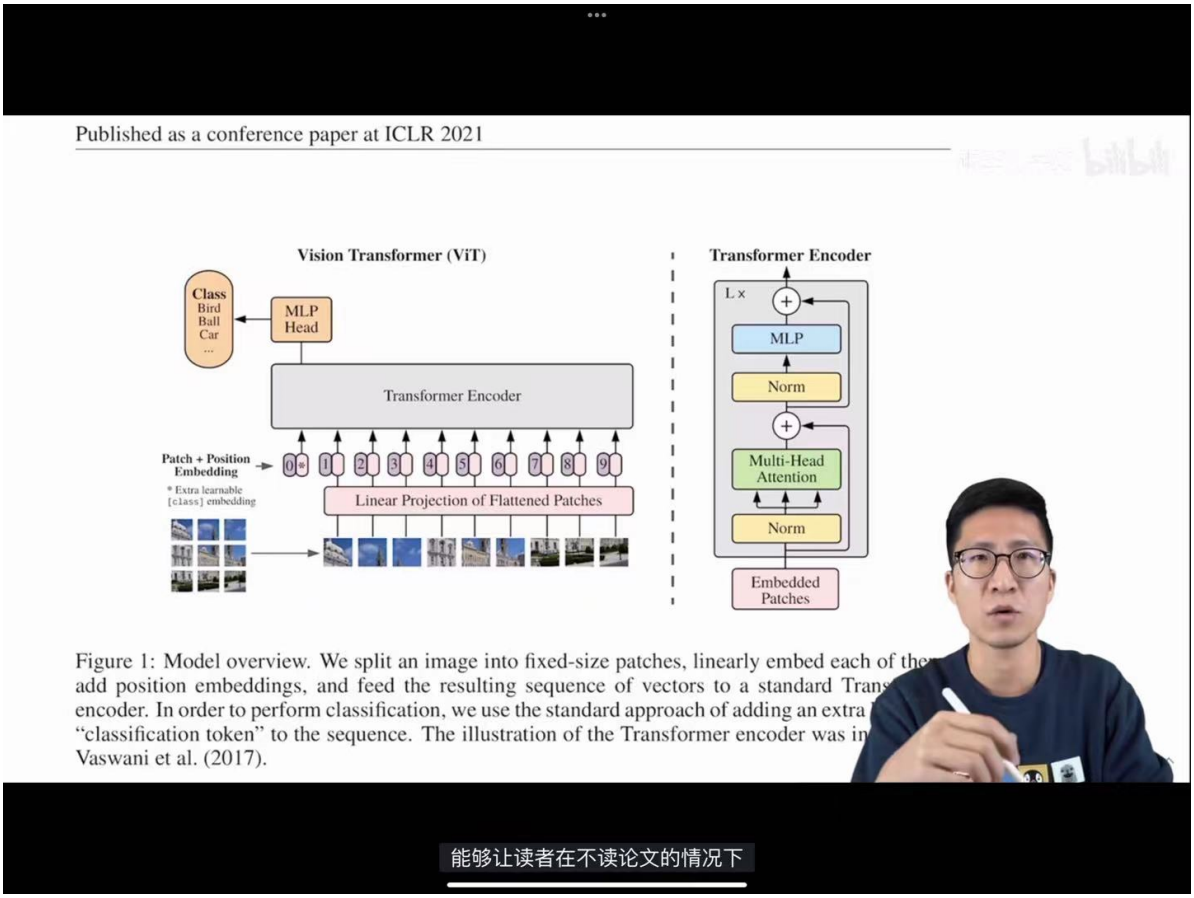
于是本文想 直接使用传统nlptransformer模型，就不会很难加速，为此将图片分为多个patch（16*16）。

vit的稳健性还是很好的

除了一开始将图像打成patches，没有再对图像进行什么 inductive biases

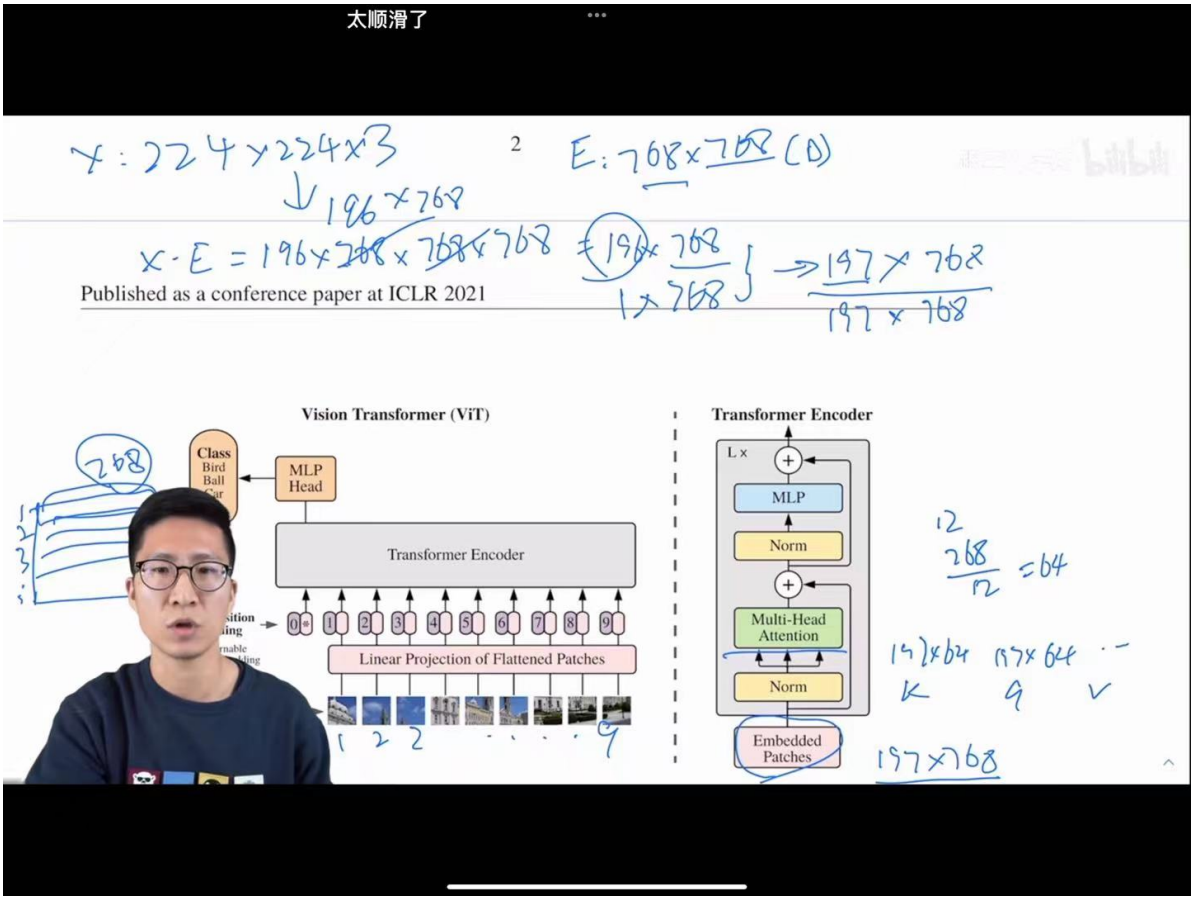
vit ； DETR;SETR; ViT-FRCNN 分别是classification、 segmentation、detection领域的模型‘

Swin-transformer ； MAE 等经典模型



输入 196个token 每个token维度是768 =16*16*3 还有一个cls字符

本文顺带还很好地讲了一下实现细节。



包括每个向量多大 怎么连接等；

注意： 这个clstoken借鉴了bert模型，是一个可以学习的token

启示 vit必须要在大规模数据集才能取得比resnet更好的效果 因此 如何使用vit去做小样本学习是一个充满潜力的方向；

感觉vit也可以用在带时间戳的数据上 甚至是视频（一点自己的不成熟的想法）

3 transformer代码带读；首先 从输入和输出层面搞懂他在干什么，代码来源

体会：可以先把大的框架搭建好，再去实现细节；第二就是一定要搞清楚每一步的数据流动方式 数据大小等；

视频作者给代码加了很详细的注释！(自己可以在复习的时候看)

2 论文（Video to Video Synthesis)

我们研究了视频到视频的合成问题，其目标是从输入源视频（例如，语义分割掩码序列）学习映射函数到输出逼真视频，精确地描述源视频的内容。虽然其图像对应的图像到图像的转换问题是一个流行的话题，但视频到视频的合成问题在文献中探讨较少。在不对时间动态进行建模的情况下，直接将现有的图像合成方法应用于输入视频，往往会导致低视觉质量的时间不相干视频。在本文中，我们提出了一种在生成对抗学习框架下的视频到视频的合成方法。通过精心设计的生成器和鉴别器，再加上时空对抗目标，我们在分割掩模、草图和姿态等不同的输入格式上获得了高分辨率、逼真的、时间一致的视频结果。在多个基准测试上的实验表明，与强基线相比，我们的方法具有优势。特别是，我们的模型能够合成2K分辨率的街景视频，最长可达30秒的时间，这大大提高了最先进的视频合成水平。最后，我们将我们的方法应用于未来的视频预测，优于几个竞争的系统。代码、模型和更多的结果可以在我们的网站上找到。

我们将视频到视频的合成问题转换为一个分布匹配问题，其目标是训练一个模型，使给定输入视频的合成视频的条件分布类似于真实视频。为此，我们学习了一个有条件生成的对抗模型

我们在几个数据集上进行了广泛的实验，将一系列分割掩模转换为逼真的视频。定量和定性的结果表明，我们的合成镜头看起来比那些从强基线更逼真。例如，请参见图1。我们进一步证明，所提出的方法可以生成逼真的2K分辨率的视频，长达30秒。我们的方法还允许用户对视频生成结果进行灵活的高级控制。例如，用户可以很容易地用街景视频中的树木替换所有的建筑物。此外，我们的方法适用于其他输入视频格式，如人脸草图和身体姿势，使许多应用程序从人脸转换到人体运动传输。最后，我们将我们的方法扩展到未来的预测，并表明我们的方法可以优于现有的系统。请访问我们的网站的代码，模型，和更多的结果。

对于image2image这个问题有大量的工作，[6,31,33,43,44,63,66,73,82,83]。我们的方法是他们的视频对应物。除了确保每个视频帧看起来逼真之外，视频合成模型还必须产生时间相干的帧，这是一项具有挑战性的任务，特别是对于一个长时间的视频。

无条件视频合成：最近的工作[59,67,69]扩展了无条件视频合成的GAN框架，它学习了一个将随机向量转换为视频的生成器。比如

VGAN [69]使用了一个时空卷积网络。TGAN [59]将一个潜在代码投射到一组潜在图像代码中，并使用一个图像生成器将这些潜在图像代码转换为帧。MoCoGAN [67]将潜在空间分解到运动子空间和内容子空间，并使用递归神经网络生成一系列运动码。由于无条件的设置，这些方法通常会产生低分辨率和短长度的视频

Future video prediction

用观测到的帧预测未来的帧，它以图像重建loss 进行优化，常常会导致模糊，同时也无法完成长时间的预测毕竟信息有限。

video-to-video合成问题与视频预测在本质上是不同的。因为video-to-video合成不会尝试预测物体的运动，而是根据已有的条件信息来产生另一个域的对应信息。

视频去噪、去模糊、去雨等也可以看作是video-to-vide synthesis问题。但这类研究都是对特定任务的，因此不能直接把已有的去噪之类的方法拿来做为本文的方法。

具体做法：

3 Video-to-Video Synthesis

Let $\mathbf{s}_1^T \equiv \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_T\}$ be a sequence of source video frames. For example, it can be a sequence of semantic segmentation masks or edge maps. Let $\mathbf{x}_1^T \equiv \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$ be the sequence of corresponding real video frames. The goal of video-to-video synthesis is to learn a mapping function that can convert $\mathbf{s}_1^T$ to a sequence of output video frames, $\tilde{\mathbf{x}}_1^T \equiv \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, ..., \tilde{\mathbf{x}}_T\}$, so that the conditional distribution of $\tilde{\mathbf{x}}_1^T$ given $\mathbf{s}_1^T$ is identical to the conditional distribution of $\mathbf{x}_1^T$ given $\mathbf{s}_1^T$.

$$p(\tilde{\mathbf{x}}_1^T|\mathbf{s}_1^T) = p(\mathbf{x}_1^T|\mathbf{s}_1^T). \tag{1}$$

Through matching the conditional video distributions, the model learns to generate photorealistic, temporally coherent output sequences as if they were captured by a video camera.

We propose a conditional GAN framework for this conditional video distribution matching task. Let G be a generator that maps an input source sequence to a corresponding output frame sequence: $\mathbf{x}_1^T = G(\mathbf{s}_1^T)$. We train the generator by solving the minimax optimization problem given by

$$\max_D \min_G E_{(\mathbf{x}_1^T, \mathbf{s}_1^T)}[\log D(\mathbf{x}_1^T, \mathbf{s}_1^T)] + E_{\mathbf{s}_1^T}[\log(1 - D(G(\mathbf{s}_1^T), \mathbf{s}_1^T))], \tag{2}$$

where D is the discriminator. We note that as solving (2), we minimize the Jensen-Shannon divergence between $p(\tilde{\mathbf{x}}_1^T|\mathbf{s}_1^T)$ and $p(\mathbf{x}_1^T|\mathbf{s}_1^T)$ as shown by Goodfellow et al. [20].

Solving the minimax optimization problem in (2) is a well-known, challenging task. Careful designs of network architectures and objective functions are essential to achieve good performance as shown in the literature [14, 21, 30, 37, 49, 51, 55, 73, 80]. We follow the same spirit and propose new network designs and a spatio-temporal objective for video-to-video synthesis as detailed below.

Sequential generator. To simplify the video-to-video synthesis problem, we make a Markov assumption where we factorize the conditional distribution $p(\tilde{\mathbf{x}}_1^T|\mathbf{s}_1^T)$ to a product form given by

$$p(\tilde{\mathbf{x}}_1^T|\mathbf{s}_1^T) = \prod_{t=1}^{T} p(\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^t). \tag{3}$$

然后所有的步骤都是为了求解这个minmax问题；

序列生成器：将条件分布写为上面那个乘积形式

In other words, we assume the video frames can be generated sequentially, and the generation of the t-th frame $\tilde{\mathbf{x}}_t$ only depends on three factors: 1) current source frame $\mathbf{s}_t$, 2) past L source frames $\mathbf{s}_{t-L}^{t-1}$, and 3) past L generated frames $\tilde{\mathbf{x}}_{t-L}^{t-1}$. We train a feed-forward network F to model the conditional

and 3) past L generated frames $\tilde{\mathbf{x}}_{t-L}^{t-1}$. We train a feed-forward network F to model the conditional distribution $p(\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^{t})$ using $\tilde{\mathbf{x}}_t = F(\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^{t})$. We obtain the final output $\tilde{\mathbf{x}}_1^T$ by applying the function F in a recursive manner. We found that a small L (e.g., $L = 1$) causes training instability, while a large L increases training time and GPU memory but with minimal quality improvement. In our experiments, we set $L = 2$.

Video signals contain a large amount of redundant information in consecutive frames. If the optical flow [46] between consecutive frames is known, we can estimate the next frame by warping the current frame [54, 70]. This estimation would be largely correct except for the occluded areas. Based on this observation, we model F as

$$F(\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^{t}) = (\mathbf{1} - \tilde{\mathbf{m}}_t) \odot \tilde{\mathbf{w}}_{t-1}(\tilde{\mathbf{x}}_{t-1}) + \tilde{\mathbf{m}}_t \odot \tilde{\mathbf{h}}_t, \tag{4}$$

where $\odot$ is the element-wise product operator and $\mathbf{1}$ is an image of all ones. The first part corresponds to pixels warped from the previous frame, while the second part hallucinates new pixels. The definitions of the other terms in Equation 4 are given below.

- $\tilde{\mathbf{w}}_{t-1} = W(\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^{t})$ is the estimated optical flow from $\tilde{\mathbf{x}}_{t-1}$ to $\tilde{\mathbf{x}}_t$, and W is the optical flow prediction network. We estimate the optical flow using both input source images $\mathbf{s}_{t-L}^{t}$ and previously synthesized images $\tilde{\mathbf{x}}_{t-L}^{t-1}$. By $\tilde{\mathbf{w}}_{t-1}(\tilde{\mathbf{x}}_{t-1})$, we warp $\tilde{\mathbf{x}}_{t-1}$ based on $\tilde{\mathbf{w}}_{t-1}$.
- $\tilde{\mathbf{h}}_t = H(\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^{t})$ is the hallucinated image, synthesized directly by the generator H.
- $\tilde{\mathbf{m}}_t = M(\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^{t})$ is the occlusion mask with continuous values between 0 and 1. M denotes the mask prediction network. Our occlusion mask is soft instead of binary to better handle the "zoom in" scenario. For example, when an object is moving closer to our camera, the object will become blurrier over time if we only warp previous frames. To increase the resolution of the object, we need to synthesize new texture details. By using a soft mask, we can add details by gradually blending the warped pixels and the newly synthesized pixels.

We use residual networks [26] for M, W, and H. To generate high-resolution videos, we adopt a

视频信号在连续的帧中包含大量的冗余信息

.M为掩模预测网络。我们的遮挡掩码是软的，而不是二进制的，以更好地处理"放大"的场景。例如，当一个物体靠近我们的相机时，如果我们只扭曲之前的帧，这个物体就会随着时间的推移而变得更加模糊。为了提高物体的分辨率，我们需要合成新的纹理细节。通过使用软掩模，我们可以通过逐步混合扭曲的像素和新合成的像素来添加细节。

Conditional image discriminator D_I. The purpose of D_I is to ensure that each output frame resembles a real image given the same source image. This conditional discriminator should output 1 for a true pair $(\mathbf{x}_t, \mathbf{s}_t)$ and 0 for a fake one $(\tilde{\mathbf{x}}_t, \mathbf{s}_t)$.

Conditional video discriminator D_V. The purpose of D_V is to ensure that consecutive output frames resemble the temporal dynamics of a real video given the same optical flow. While D_I conditions on the source image, D_V conditions on the flow. Let $\mathbf{w}_{t-K}^{t-2}$ be $K-1$ optical flow for the K consecutive real images $\mathbf{x}_{t-K}^{t-1}$. This conditional discriminator D_V should output 1 for a true pair $(\mathbf{x}_{t-K}^{t-1}, \mathbf{w}_{t-K}^{t-2})$ and 0 for a fake one $(\tilde{\mathbf{x}}_{t-K}^{t-1}, \mathbf{w}_{t-K}^{t-2})$.

While *DI *conditions on the source image, *DV* conditions on the flow.

Sequential generator

为了简化问题，作出Markov assumption，将条件概率$\mathrm{p}\left(\tilde{\mathbf{x}}_1^{\mathrm{T}} \mid \mathbf{s}_1^{\mathrm{T}}\right)$分解为

$$\mathrm{p}\left(\tilde{\mathbf{x}}_1^{\mathrm{T}} \mid \mathbf{s}_1^{\mathrm{T}}\right)=\prod_{\mathrm{t}=1}^{\mathrm{T}} \mathrm{p}\left(\tilde{\mathbf{x}}_{\mathrm{t}} \mid \tilde{\mathbf{x}}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}-1}, \mathbf{s}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}}\right) \qquad (3)$$

上述式子的意思是，假设我们已经生成了前$\mathrm{t}-1$帧$\tilde{\mathbf{x}}_1^{\mathrm{t}-1}$，当前需要生成第$\mathrm{t}$帧$\tilde{\mathbf{x}}_{\mathrm{t}}$，使用的信息有

1. current source frame $\mathbf{s}_{\mathrm{t}}$
2. past L source frames $\mathbf{s}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}-1}$
3. past L generated frames $\tilde{\mathbf{x}}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}-1}$

其中1和2可以合并为$\mathbf{s}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}}$，$\mathrm{L}$是一个超参数，取值小会造成训练不稳定，取值大会增大GPU消耗，因此在实验中设置$\mathrm{L}=2$比较合适

将公式(3)中的条件概率$\mathrm{p}\left(\tilde{\mathbf{x}}_{\mathrm{t}} \mid \tilde{\mathbf{x}}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}-1}, \mathbf{s}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}}\right)$表达为网络$\mathrm{F}$，$\tilde{\mathbf{x}}_{\mathrm{t}}=\mathrm{F}\left(\tilde{\mathbf{x}}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}-1}, \mathbf{s}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}}\right)$，于是就可以利用网络$\mathrm{F}$逐帧地生成视频

在视频中前后帧之间往往是高度相似的，因此考虑使用光流法，如果前后两帧之间的光流已知，那么可以通过warping前一帧来生成下一帧

具体来说，网络F表达为

$$\mathrm{F}\left(\tilde{\mathbf{x}}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}-1}, \mathbf{s}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}}\right)=\left(1-\tilde{\mathbf{m}}_{\mathrm{t}}\right) \odot \tilde{\mathbf{w}}_{\mathrm{t}-1}\left(\tilde{\mathbf{x}}_{\mathrm{t}-1}\right)+\tilde{\mathbf{m}}_{\mathrm{t}} \odot \tilde{\mathbf{h}}_{\mathrm{t}} \qquad (4)$$

第1项为光流法中warp前一帧的结果，第2项为生成的图像（hallucinates new pixels），二者使用mask $\tilde{\mathbf{m}}_{\mathrm{t}}$做权衡
Q：$\tilde{\mathbf{w}}_{\mathrm{t}-1}\left(\tilde{\mathbf{x}}_{\mathrm{t}-1}\right)$对应了warp操作，可以直接理解为矩阵乘法吗

网络F的运算中又涉及到了3个网络W，H和M

- $\tilde{\mathbf{x}}_{\mathrm{t}-1}=\mathrm{W}\left(\tilde{\mathbf{x}}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}-1}, \mathbf{s}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}}\right)$表示从帧$\tilde{\mathbf{x}}_{\mathrm{t}-1}$到$\tilde{\mathbf{x}}_{\mathrm{t}}$，使用optical flow prediction network W 预测的光流
- $\tilde{\mathbf{h}}_{\mathrm{t}}=\mathrm{H}\left(\tilde{\mathbf{x}}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}-1}, \mathbf{s}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}}\right)$表示由generator H生成的hallucinated image
- $\tilde{\mathbf{m}}_{\mathrm{t}}=\mathrm{M}\left(\tilde{\mathbf{x}}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}-1}, \mathbf{s}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}}\right)$表示由mask prediction network M生成的occlusion mask，未被遮挡的部分可以使用光流解决，被遮挡的部分只能从$\tilde{\mathbf{h}}_{\mathrm{t}}$中获取

训练网络F的时候，必须采用coarse-to-fine的方式

【关于光流】
光流定义为图像中的像素的运动速度，前后两帧之间的光流需要使用特定算法（Gunner Farneback's algorithm）来计算，在本文中使用FlowNet2来计算

Conditional image discriminator

定义image级别的conditional 判别器D_{I}，用于判别真实的pair $\left(\mathbf{x}_{\mathrm{t}}, \mathbf{s}_{\mathrm{t}}\right)$和假的pair $\left(\tilde{\mathbf{x}}_{\mathrm{t}}, \mathbf{s}_{\mathrm{t}}\right)$

Conditional video discriminator

定义video级别的conditional 判别器D_{V}，给定光流作为条件，判别真假output frames

具体来说，对于连续的K个real images $\mathbf{x}_{\mathrm{t}-\mathrm{K}}^{\mathrm{t}-1}$，其光流序列为$\mathbf{w}_{\mathrm{t}-\mathrm{K}}^{\mathrm{t}-2}$，那么$\mathrm{D}_{\mathrm{V}}$负责判别真实的pair $\left(\mathbf{x}_{\mathrm{t}-\mathrm{K}}^{\mathrm{t}-1}, \mathbf{w}_{\mathrm{t}-\mathrm{K}}^{\mathrm{t}-2}\right)$和假的pair $\left(\tilde{\mathbf{x}}_{\mathrm{t}-\mathrm{K}}^{\mathrm{t}-1}, \mathbf{w}_{\mathrm{t}-\mathrm{K}}^{\mathrm{t}-2}\right)$

Foreground-background prior

当使用semantic segmentation masks作为source video时，可以将semantic segmentation分为foreground和background，利用这个信息可以生成更好的video

具体来说，将image hallucination network H拆分为foreground model🔍 $\tilde{\mathbf{h}}_{\mathrm{F},\mathrm{t}}=\mathrm{H}_{\mathrm{F}}\left(\mathbf{s}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}}\right)$和background model $\tilde{\mathbf{h}}_{\mathrm{B},\mathrm{t}}=\mathrm{H}_{\mathrm{B}}\left(\tilde{\mathbf{x}}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}-1}, \mathbf{s}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}}\right)$，则公式(4)修改如下

$$\mathrm{F}\left(\tilde{\mathbf{x}}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}-1}, \mathbf{s}_{\mathrm{t}-\mathrm{L}}^{\mathrm{t}}\right)=\left(1-\tilde{\mathbf{m}}_{\mathrm{t}}\right) \odot \tilde{\mathbf{w}}_{\mathrm{t}-1}\left(\tilde{\mathbf{x}}_{\mathrm{t}-1}\right)+\tilde{\mathbf{m}}_{\mathrm{t}} \odot\left(\left(1-\mathbf{m}_{\mathrm{B},\mathrm{t}}\right) \odot \tilde{\mathbf{h}}_{\mathrm{F},\mathrm{t}}+\mathbf{m}_{\mathrm{B},\mathrm{t}} \odot \tilde{\mathbf{h}}_{\mathrm{B},\mathrm{t}}\right) \qquad (9)$$

其中$\mathbf{m}_{\mathrm{B},\mathrm{t}}$是根据ground truth segmentation mask $\mathbf{s}_{\mathrm{t}}$计算得到的

使用Foreground-background prior可以极大地提高生成video的视觉质量，付出的代价仅仅是video中会有一些轻微的闪烁

Multimodal synthesis

在特征空间上做一些随机处理，从而可以生成多段不同的视频

4 Experiments

总共进行了3种类型的视频生成

总共进行了3种类型的视频生成

1. Semantic manipulation，见Figure 2
2. Sketch-to-video synthesis for face swapping，见Figure 5
3. Pose-to-video synthesis for human motion transfer，见Figure 6

multimodel synthesis： 合成网络F是一个单模型映射函数。给定一个输入源视频，它只能生成一个输出视频。为了实现多模态合成[19,73,83]，我们对一个由实例级语义分割掩码组成的源视频采用了特征嵌入方案[73]。具体来说，在训练时，我们训练一个图像编码器E将完全真实图像xt编码为d维特征图（在我们的实验中是d=3）。然后，我们对映射应用一个实例级的平均池化，从而使同一对象中的所有像素共享相同的特征向量。然后，我们将实例平均特征映射zt和输入语义分割掩模st输入给生成器f。一旦训练完成，我们将高斯分布的混合物拟合到属于同一对象类的特征向量上。在测试时，我们使用每个对象类的估计分布来为每个对象实例采样一个特征向量。给定不同的特征向量，生成器F可以合成具有不同视觉外观的视频。

【总结】

本文提出了Video-to-Video生成的方法，相当于将pix2pix扩展到video上，在训练时，需要使用逐帧对应的两个视频序列（semantic segmentation mask -> video，sketch -> video，pose -> video）进行训练，在测试时，以dance video为例，给定一段video，对其提取pose序列，然后可以生成一段逼真的video，相当于将视频中的人进行了替换

建议搞一下这个的代码。该模型效果尚可