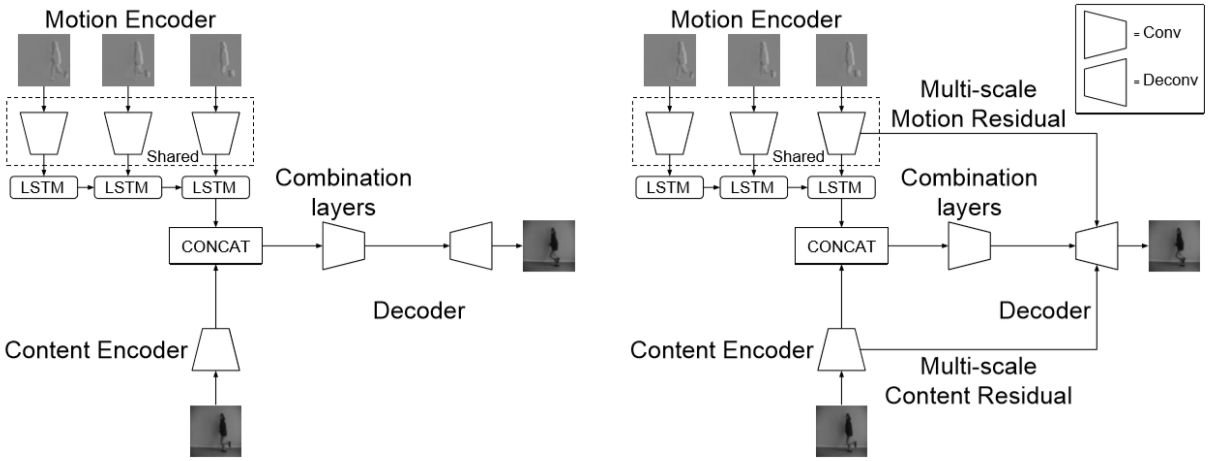
MCnet简介

本文的思路主要是用两个不同的网络把motion和content分开。

模型简介

其中motion的输入是序列之差，content的输入是最后一帧图像。motion 采用convlstm的方式进行编码，content采用CNN方式编码。为了提高生成质量，还采用了Unet的残差结构。



(a) Base MCnet (b) MCnet with Multi-scale Motion-Content Residuals

- content encoder用了VGG16直到第三个pooling层的结构。
- motion encoder也是VGG16 3rd pooling。只是把连续的3x3卷积换成了5x5, 5x5和7x7的卷积结构。
- comb layer用了三个3x3 conv(256, 128和256 channels)。multi-scale residuals是两个3x3卷积。
- decoder结构和content encoder刚好相反，用deconv和unpool，

而baseline convlstm的结构和motion encoder,res connection和decoder一样，除了encoder的channel有所增大和MCnet的总参数数量差不多。

算法框架

motion encoder

$$[\mathbf{d}_t, \mathbf{c}_t] = f^{\text{dyn}}\left(\mathbf{x}_t - \mathbf{x}_{t-1}, \mathbf{d}_{t-1}, \mathbf{c}_{t-1}\right),$$

d,c可以认为是conv lstm的cell state和hidden state，先用cnn encode然后再接convlstm的结构，是纯conv结构。

content encoder

$$\mathbf{s}_t = f^{\text{cont}}(\mathbf{x}_t),$$

用cnn encode

multi scale motion-content residual

$$\mathbf{r}_t^l = f^{\text{res}}\left(\left[\mathbf{s}_t^l, \mathbf{d}_t^l\right]\right)^l,$$

将motion和content encoder第l层的输出concate后送入res模块，由连续conv和relu组成

comb layer and decoder

$$\mathbf{f}_t = g^{\text{comb}}\left(\left[\mathbf{d}_t, \mathbf{s}_t\right]\right),$$

首先是把content和motion的feature结合在一起然后送一个cnn bottlenect layer将d_t和s_t映射到一个低维度的embedding space然后再搞回同一尺寸这里每太看懂。然后可认为f_t就是下一帧的content feature S_t+1。最后是decoder的结构

$$\hat{\mathbf{x}}_{t+1} = g^{\text{dec}}\left(\mathbf{f}_t, \mathbf{r}_t\right),$$

问题

1. residual的结构到底是怎样，不是简单的unet res么为什么有3x3 conv
2. deconv, unpool，棋盘效应?
3. comb layer why lower dimension