Video Matting with Convolutional LSTM

Jiahao Zhang U6921098 | Peng Zhang U6921163 | Hang Zhang U6921112

Abstract

TODO

1. Introduction

background TODO
motivation TODO
related works TODO
Here summaries our main contributions,

- To our best knowledge, it is the first time to combinate ConvLSTM with Deeplab architect for video matting.
- We persent a possibility to apply video matting without background as input.

2. Problem Statement

definition, and formulation TODO

3. Methods

In this section, we will introduce both model structure and loss functions for this project.

3.1. Model Structure

The model is inspired from the base part of [2], which can be divided into three parts as shown in Figure 1. One of the major differences is the input size. Instead of images with background ($B \times 6 \times H \times W$), we use video frames without background ($B \times T \times 3 \times H \times W$) where T refers to time. Another is we add ConvLSTM after each Decoder Block to obtain temporal features.

3.1.1 Encoder

The Encoder consists of four encoder blocks (EB), each contains a convolutional layer, a batch normalization layer and a ReLU activation layer. This architecture is taken from ResNet50 pretrained on ImageNet. It is by design that after each EB, the size of feature maps halves, and the number of channels increases in the order $[3, 64, 256, 512, 2048]$.

3.1.2 ASPP

Atrous Spatial Pyramid Pooling (ASPP) utilizes the fusion of multiple convolutions with different dilation rates to increase receptive field, shrink the size of feature maps, just like pooling. But, meanwhile, it can keep more informative features.

3.1.3 Decoder

The decoder can be divided into three decoder blocks (DB) and an output block (OB), each of them starts with upsampling, implemented by 2x bilinear interpolation. Together with output of each EB, the feature map then pass a convolutional layer, a batch normalization layer, a ReLU activation layer and ConvLSTM layer. The output size of each DB doubles and the number of channels decreases in the order $[768, 384, 128, 51, 37]$. The OB generates the output directly, which contains 37 channels. The first channel is Alpha, which is the matte in greyscale. The next 3 channels are Foreground, it is designed to make the model focus more on the foreground. The next channel is Error, it is the predicted error, which will then be compared with the error between the predicted alpha and the ground truth alpha. The rest channels are set for the refine part of [2], which is not used in this project. But we still remain these channels for easier transfer learning.

3.1.4 ConvLSTM

$$
\begin{aligned}
\mathbf{i_t} &= \text{Sigmoid}\left(\text{Conv}\left(\mathbf{x_t}; \mathbf{w_{xi}}\right) + \text{Conv}\left(\mathbf{h_{t-1}}; \mathbf{w_{hi}}\right) + \mathbf{b_i}\right) \\
\mathbf{f_t} &= \text{Sigmoid}\left(\text{Conv}\left(\mathbf{x_t}; \mathbf{w_{xf}}\right) + \text{Conv}\left(\mathbf{h_{t-1}}; \mathbf{w_{hf}}\right) + \mathbf{b_f}\right) \\
\mathbf{o_t} &= \text{Sigmoid}\left(\text{Conv}\left(\mathbf{x_t}; \mathbf{w_{xo}}\right) + \text{Conv}\left(\mathbf{h_{t-1}}; \mathbf{w_{ho}}\right) + \mathbf{b_o}\right) \\
\mathbf{g_t} &= \text{Tanh} \quad \left(\text{Conv}\left(\mathbf{x_t}; \mathbf{w_{xg}}\right) + \text{Conv}\left(\mathbf{h_{t-1}}; \mathbf{w_{hg}}\right) + \mathbf{b_g}\right) \\
\mathbf{c_t} &= \mathbf{f_t} \odot \mathbf{c_{t-1}} + \mathbf{i_t} \odot \mathbf{g_t} \\
\mathbf{h_t} &= \mathbf{o_t} \odot \text{Tanh}\left(\mathbf{c_t}\right)
\end{aligned}
\tag{1}
$$

3.2. Loss Function

TODO

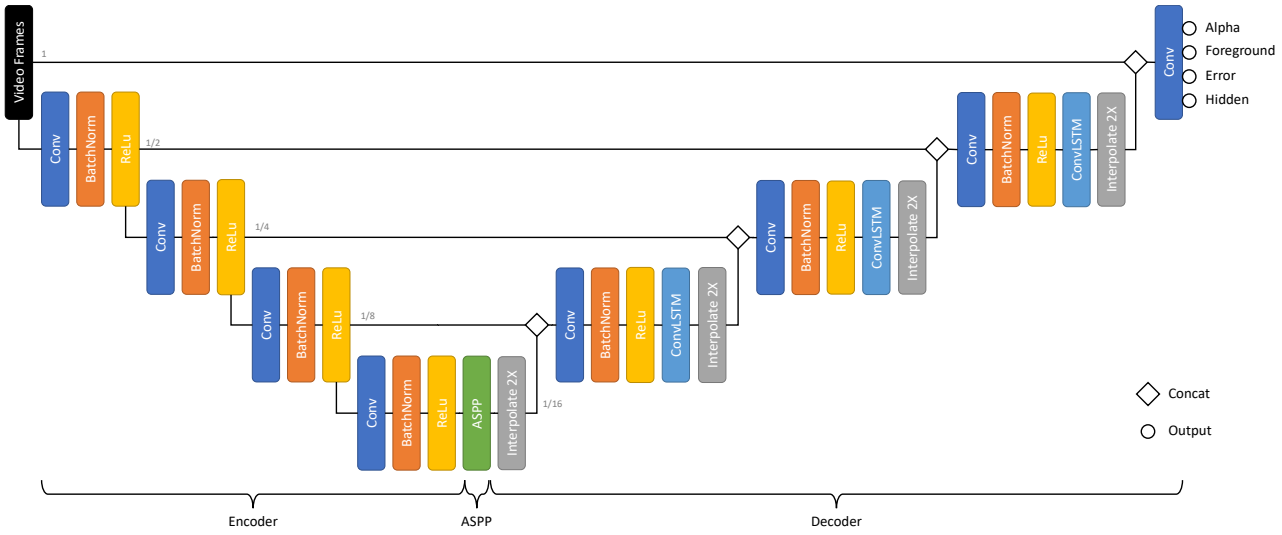

Figure 1. The architecture of our network.

4. Experiments

TODO

4.1. Experiment Setup

4.1.1 Datasets

TODO

4.1.2 Metrics

TODO

MAD (mean absolute difference) TODO

MSE (mean squared error) TODO

GRAD (Gradient) TODO

CONN (Connectivity) TODO

4.1.3 Implementation

TODO

4.2. Experiment Results

TODO

4.2.1 Comparing

TODO

Table 1. caption

	MAD	MSE	GRAD	CONN
original	**2.04**	**0.94**	**0.11**	**102.40**
transfer	8.09	5.44	6.73	406.09
convLSTM	5.18	3.73	7.65	259.67

4.2.2 Ablation Study

TODO

5. Conclusion

Conclusion TODO
[1]

6. References

7. References

[1] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv:1706.05587 [cs]*, Dec. 2017. arXiv: 1706.05587. 2

[2] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian Curless, Steve Seitz, and Ira Kemelmacher-Shlizerman. Real-Time High-Resolution Background Matting. *arXiv:2012.07810 [cs]*, Dec. 2020. arXiv: 2012.07810 version: 1. 1

8. Review

8.1. Self Reflection

TODO

8.2. Confidential Peer Review

TODO

In doing this project, to the best of my judgement, I confirm that Jiahao Zhang mainly contributed to TODO, and his/her overall contribution is about 34%, Peng Zhang mainly worked on TODO, and his/her contribution is about 33%, and Hang Zhang was responsible for TODO, and his/her contribution counts about 33% of the total project workload.