A Flexible Recurrent Residual Pyramid Network for Video Frame Interpolation Supplementary Material

Anonymous ECCV submission

Paper ID 5095

In this supplementary document, we present additional results to complement the paper. Section A summarizes the details of our network. Section B provides more visual comparison results on various resolution cases. More video results are also provided on Supplementary Material.

A Network Details

The configuration details of the Recurrent Residual Layer (RRL) are provided in Table 1. Batch normalization is adopted on RRL for accelerating convergence. The average pooling layers and bilinear upsampling layers are used to decrease and increase the spatial dimension by a factor of 2 for encoder and decoder, respectively. The α of Leaky ReLU is set to be 0.1. In our refinement network, we use 3 residual blocks to predict the residual between the blended warped frames and the ground-truth frame. Table 2 provides the configuration details of the refinement network.

Table 1. Detailed configuration of the Recurrent Residual Layer.

	Input	Output	Kernel size	#input channels	#output channels	Stride	Activation	Output size
feature extractor	RGB1 (RGB2)	fea1_conv1 (fea2_conv1)	3×3	3	16	1	Leaky ReLU	$H \times W$
	fea1_conv1 (fea2_conv1)	fea1_conv2 (fea2_conv2)	3×3	16	16	1	Leaky ReLU	$H \times W$
	fea1_conv2 (fea2_conv2)	fea1_conv3 (fea2_conv3)	3×3	16	16	1	-	$H \times W$
encoder	features	enc1_conv1	7×7	38	32	1	Leaky ReLU	$H \times W$
	enc1_conv1	enc1_conv2	7×7	32	32	1	Leaky ReLU	$H \times W$
	enc1_conv2	enc_pool1	2×2	32	32	2	-	$H/2 \times W/2$
	enc_pool1	enc2_conv1	3×3	32	64	1	Leaky ReLU	$H/2 \times W/2$
	enc2_conv1	enc2_conv2	3×3	64	64	1	Leaky ReLU	$H/2 \times W/2$
	enc2_conv2	enc_pool2	2×2	64	64	2	-	$H/4 \times W/4$
	enc_pool2	enc3_conv1	3×3	64	128	1	Leaky ReLU	$H/4 \times W/4$
	enc3_conv1	enc3_conv2	3×3	128	128	1	Leaky ReLU	$H/4 \times W/4$
	enc3_conv2	enc_pool3	2×2	128	128	2	-	$H/8 \times W/8$
	enc_pool3	enc4_conv1	3×3	128	256	1	Leaky ReLU	$H/8 \times W/8$
	enc4_conv1	enc4_conv2	3×3	256	256	1	Leaky ReLU	$H/8 \times W/8$
	enc4_conv2	enc_pool4	2×2	256	256	2	-	$H/16 \times W/16$
	enc_pool4	enc5_conv1	3×3	256	256	1	Leaky ReLU	$H/16 \times W/16$
	enc5_conv1	enc5_conv2	3×3	256	256	1	Leaky ReLU	$H/16 \times W/16$
decoder	enc5_conv2(up)	dec1_conv1	3×3	256	256	1	Leaky ReLU	$H/8 \times W/8$
	dec1_conv1+enc4_conv2	dec1_conv2	3×3	256+256	256	1	Leaky ReLU	$H/8 \times W/8$
	dec1_conv2(up)	dec2_conv1	3×3	256	128	1	Leaky ReLU	$H/4 \times W/4$
	dec2_conv1+enc3_conv2	dec2_conv2	3×3	128+128	128	1	Leaky ReLU	$H/4 \times W/4$
	dec2_conv2(up)	dec3_conv1	3×3	128	64	1	Leaky ReLU	$H/2 \times W/2$
	dec3_conv1+enc2_conv2	dec3_conv2	3×3	64+64	64	1	Leaky ReLU	$H/2 \times W/2$
	dec3_conv2(up)	dec4_conv1	3×3	64	32	1	Leaky ReLU	$H \times W$
	dec4_conv1+enc1_conv2	dec4_conv2	3×3	32+32	32	1	Leaky ReLU	$H \times W$
context network	dec4_conv2	out1	3×3	32	6	1	-	$H \times W$
	dec4_conv2+out1	dilat_conv1	3×3	38	32	1	Leaky ReLU	$H \times W$
	dilat_conv1	dilat_conv2	3×3	32	32	1	Leaky ReLU	$H \times W$
	dilat_conv2	dilat_conv3	3×3	32	32	1	Leaky ReLU	$H \times W$
	dilat_conv3	out2	3×3	32	6	1	-	$H \times W$

Table 2. Detailed configuration of the refinement network.

	Input	Output	Kernel size	#input channels	#output channels	Stride	Activation	Output size
in	features	in_conv	5×5	142	64	1	ReLU	$H \times W$
resblock1	in_conv	res1_conv1	7×7	64	64	1	ReLU	$H \times W$
	res1_conv1	res1_conv2	7×7	64	64	1	-	$H \times W$
	in_conv+res1_conv2	resblock1	-	64	64	1	ReLU	$H \times W$
resblock2	resblock1	res2_conv1	7×7	64	64	1	ReLU	$H \times W$
	res2_conv1	res2_conv2	7×7	64	64	1	-	$H \times W$
	resblock1+res2_conv2	resblock2	-	64	64	1	ReLU	$H \times W$
resblock3	resblock2	res3_conv1	7×7	64	64	1	ReLU	$H \times W$
	res3_conv1	res3_conv2	7×7	64	64	1	-	$H \times W$
	resblock2+res3_conv2	resblock3	-	64	64	1	ReLU	$H \times W$
out	resblock3	out_conv	7×7	64	3	1	-	$H \times W$

B Visual Comparisons

More visual comparison results on various resolution cases are provided in this section. Figure 1 and 2 shows more challenge cases on the H.266 test sequences (4K) and the ActivityNet (1080P videos). Figure 3 shows more visual comparisons on the Thumos15 (720P).

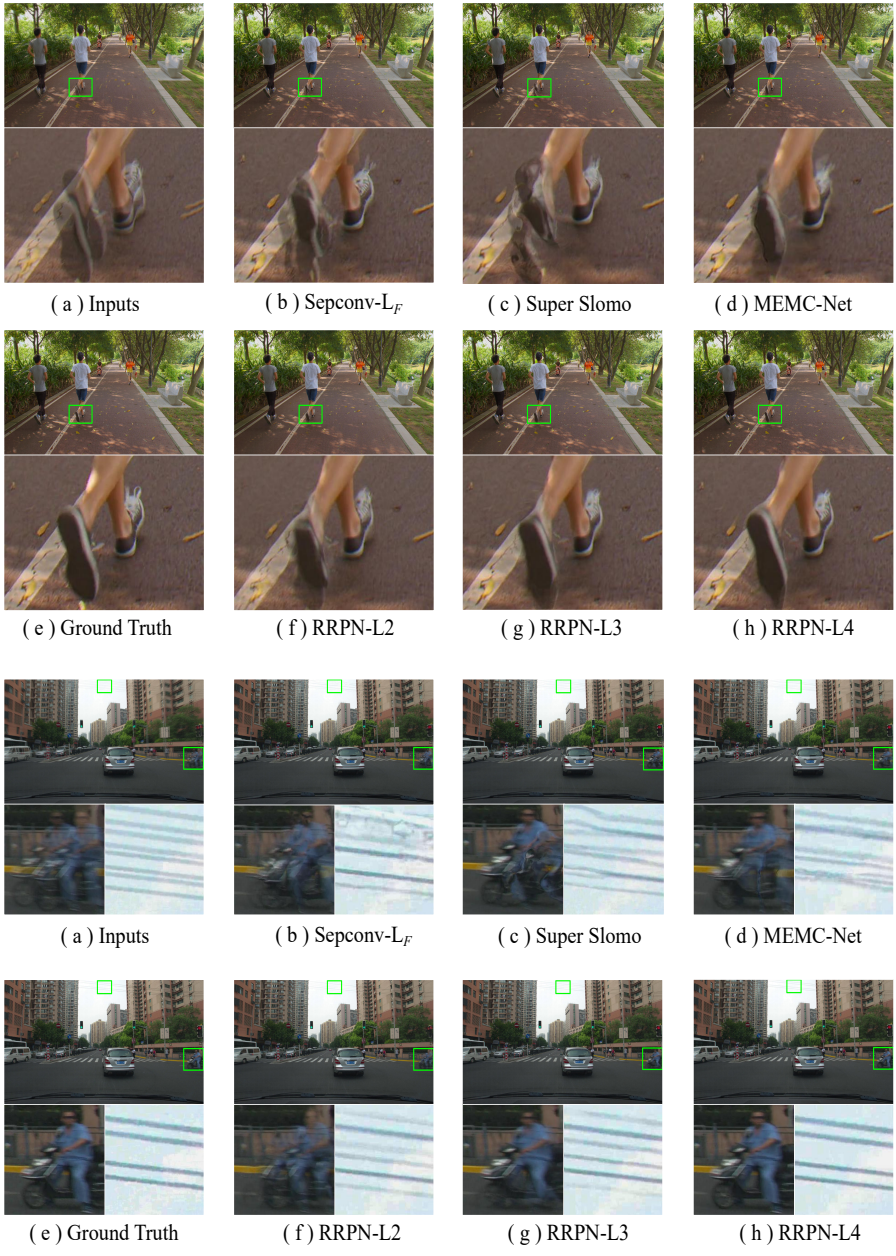

Fig. 1. More interpolation resolution from H.266 (**4K**) test data. Our approach can better capture large motions in different cases, even for small and thin objects. The contrast of image patches that contain wires are enhanced to better observe interpolation results of small and thin objects generated by different networks.

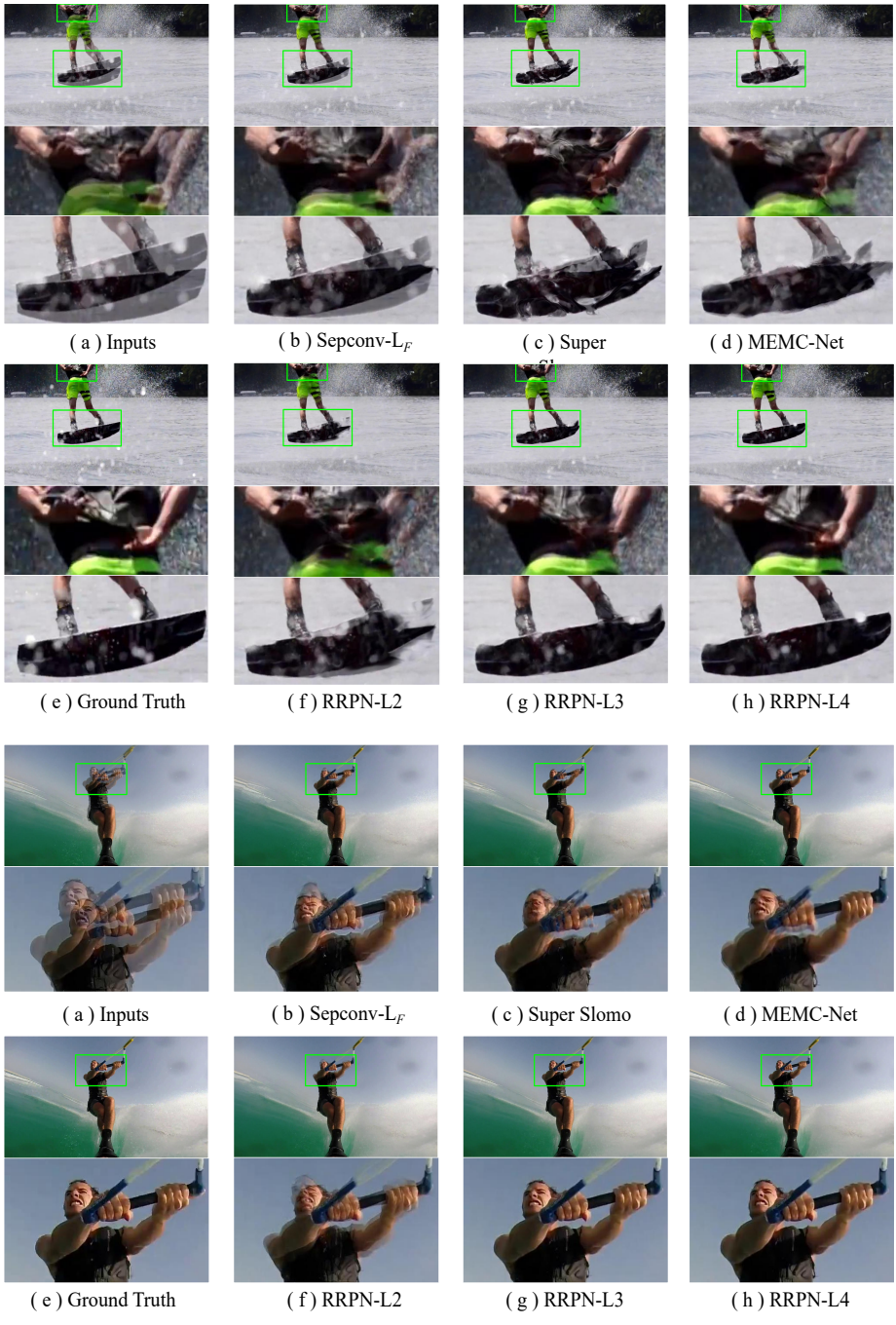

Fig. 2. More interpolation resolution from ActivityNet (**1080P**).

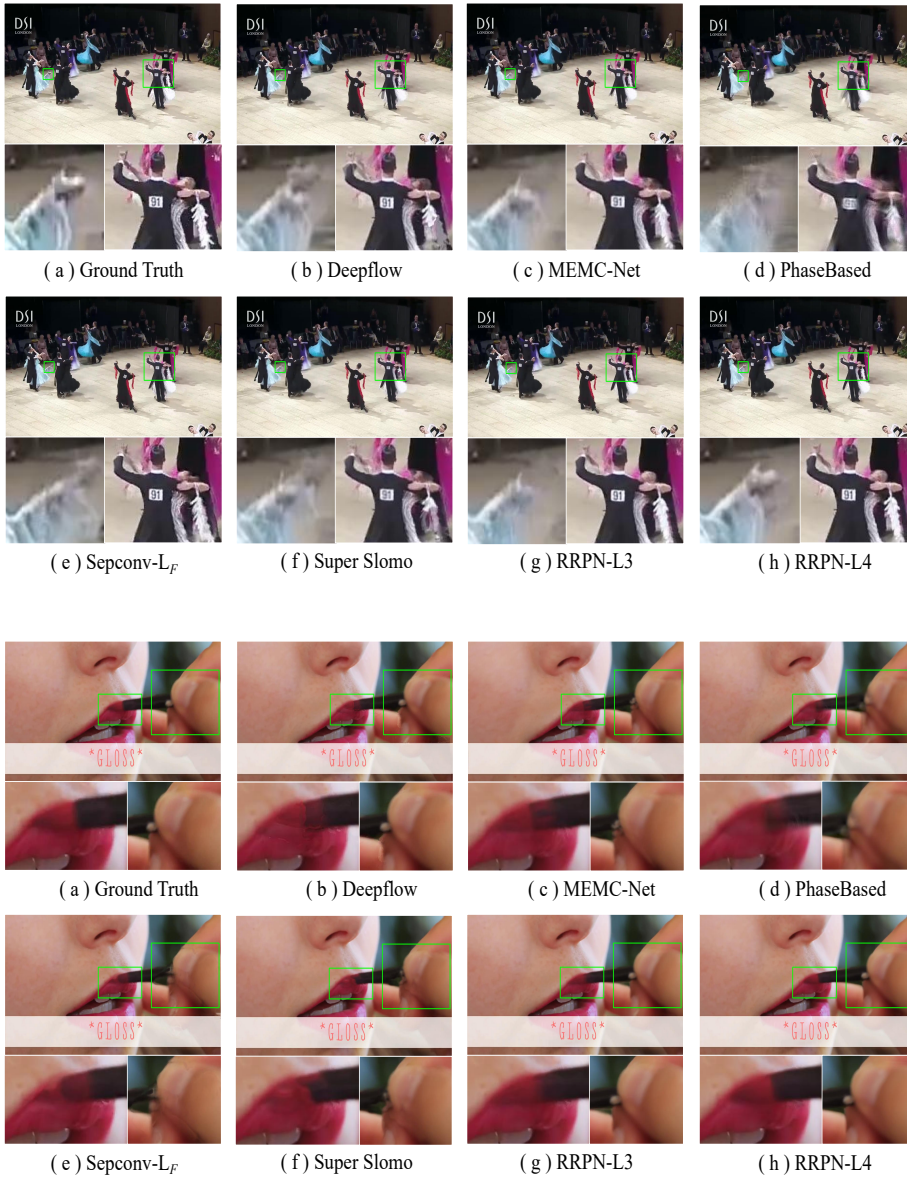

Fig. 3. More interpolation resolution from Thumos15 (**720P**).