Accelerating Video Super Resolution for Mobile Device

CSC 591/791-025 Final Project

Brian Park, Oliver Fowler

Outline

Video Super Resolution Task

Super Resolution Model Choice

Optimizations

Experimental Results

Conclusion

Video Super Resolution Task

- SOTA Super Resolution models are very large
- Tradeoff between quality and speed
- Strict latency requirements for real time video

FPS	Latency (ms)
24	41.67
30	33.33
60	16.67

DNN Model Choice

- Find a good baseline model
  - A model that achieves good baseline performance on cloud and compatible with XGen
- Reported any XGen issues we came across
- NTIRE 2022 Challenge on Efficient Super Resolution (CVPR 2022)

Team	Main Track	Sub-Track 1	Sub-Track 2	PSNR [Val.]	PSNR [Test]	Ave. Time [ms]	#Params [M]	FLOPs [G]	#Acts [M]	GPU Mem. [M]	#Conv
ByteESR	1	22(11)	33(2)	29.00	28.72	27.11(1)	0.317(11)	19.70(11)	80.05(6)	377.91(4)	39
NJU_Jet	2	37(18)	44(6)	29.00	28.69	28.07(2)	0.341(18)	22.28(19)	72.09(4)	204.60(1)	34
NEESR	3	10(4)	27(1)	29.01	28.71	29.97(3)	0.272(4)	16.86(6)	79.59(5)	575.99(9)	59
Super	4	26(12)	55(10)	29.00	28.71	32.09(4)	0.326(14)	20.06(12)	93.82(10)	663.07(15)	59
MegSR	5	18(9)	43(5)	29.00	28.68	32.59(5)	0.290(9)	17.70(9)	91.72(8)	640.63(12)	64
rainbow	6	16(8)	34(3)	29.01	28.74	34.10(6)	0.276(6)	17.98(10)	92.80(9)	309.23(3)	59
VMCL_Taobao	7	29(14)	57(11)	29.01	28.68	34.24(7)	0.323(13)	20.97(16)	98.67(11)	633.00(10)	40
Bilibili AI	8	15(7)	41(4)	29.00	28.70	34.67(8)	0.283(8)	17.61(7)	90.50(7)	633.74(11)	64
NKU-ESR	9	12(5)	48(7)	29.00	28.66	34.81(9)	0.276(7)	16.73(5)	111.12(13)	662.51(14)	65
NJUST_RESTORARION	10	54(27)	89(15)	28.99	28.68	35.76(10)	0.421(28)	27.67(26)	108.66(12)	643.95(13)	52
TOVBU	11	43(21)	96(19)	29.00	28.71	38.32(11)	0.376(23)	22.38(20)	113.55(15)	867.17(27)	64
Alpan Team	12	18(10)	51(9)	29.01	28.75	39.63(12)	0.326(15)	12.31(3)	115.52(16)	439.37(5)	132
Dragon	13	38(19)	70(13)	29.01	28.69	41.80(13)	0.358(20)	21.11(18)	120.15(17)	260.00(2)	131
TieGuoDun Team	14	54(27)	104(21)	28.95	28.65	42.35(14)	0.433(29)	27.10(25)	112.03(14)	788.13(22)	64
HiImageTeam	15	7(3)	70(13)	29.00	28.72	47.75(15)	0.242(3)	14.51(4)	151.36(23)	861.84(25)	100
xilinxSR	16	66(34)	107(22)	29.05	28.75	48.20(16)	0.790(34)	51.76(32)	136.31(18)	471.37(7)	38
cipher	17	50(24)	111(23)	29.00	28.72	51.42(17)	0.407(26)	25.25(24)	155.35(24)	770.82(20)	67
NJU_MCG	18	13(6)	66(12)	28.99	28.71	52.02(18)	0.275(5)	17.65(8)	212.35(27)	511.08(8)	84
IMGWLH	19	34(17)	91(17)	29.01	28.72	56.34(19)	0.362(21)	20.10(13)	136.35(19)	753.02(19)	113
imglhl	20	45(22)	92(18)	29.03	28.75	56.88(20)	0.381(24)	23.26(21)	144.05(21)	451.21(6)	127
whu_sigma	21	63(32)	132(30)	29.02	28.73	61.04(21)	0.705(33)	43.88(30)	142.91(20)	1011.54(28)	64
Aselsan Research	22	27(13)	98(20)	29.02	28.73	63.18(22)	0.317(12)	20.71(15)	206.05(26)	799.52(23)	134
Drinktea	23	59(31)	121(27)	29.00	28.70	75.52(23)	0.589(31)	36.92(28)	148.05(22)	734.54(17)	67
GDUT_SR	24	50(24)	136(31)	29.05	28.75	75.70(24)	0.414(27)	24.80(23)	260.05(28)	1457.98(34)	195
Giantpandacv	25	63(32)	150(34)	29.07	28.76	87.87(25)	0.683(32)	45.07(31)	361.23(31)	1272.95(31)	122
neptune	26	39(20)	123(29)	28.99	28.69	101.69(26)	0.316(10)	38.03(29)	269.48(29)	1179.05(29)	45
XPixel	27	3(1)	49(8)	29.01	28.69	140.47(27)	0.156(1)	9.50(2)	65.76(3)	729.94(16)	43
NJUST_ESR	28	3(1)	89(15)	28.96	28.68	164.80(28)	0.176(2)	8.73(1)	160.43(25)	1346.74(33)	25
TeamInception	29	57(30)	146(33)	29.12	28.82	171.56(29)	0.505(30)	32.42(27)	502.27(34)	866.16(26)	74
cceNBgdd	30	33(16)	114(24)	28.97	28.67	180.60(30)	0.339(16)	21.11(17)	404.16(33)	739.65(18)	197
ZLZ	31	55(29)	118(26)	29.00	28.72	183.43(31)	0.372(22)	64.45(33)	57.51(2)	1244.23(30)	16
Express	32	31(15)	117(25)	29.04	28.77	203.16(32)	0.339(17)	20.41(14)	325.53(30)	853.27(24)	148
Just Try	33	70(35)	170(35)	29.12	28.81	247.90(33)	0.832(35)	135.30(35)	392.43(32)	2387.93(35)	207
ncepu_explorers	34	47(23)	137(32)	29.09	28.79	317.66(34)	0.390(25)	23.73(22)	994.25(35)	771.54(21)	374
mju_mnu	35	53(26)	121(27)	29.06	28.79	332.28(35)	0.345(19)	78.81(34)	46.76(1)	1310.72(32)	40
The following methods are not ranked since their validation/testing PSNR are not on par with the baseline.											
Virtual_Reality Team				27.35	27.26	2231.32	0.423	423.16	2731.08	3336.88*	82
NTU607QCO-ESR				27.79	27.61	38.85	0.433	27.06	108.89	776.38	60
Strong Tiger				29.00	28.61	34.92	0.560	36.64	78.91	641.13	23
VAP				29.01	28.47	23.96	0.175	10.83	70.93	507.64	63
Multicog											
Set5baby Team											
NWPU_SweetDreamLab											
SSL											
RFDN AIM2020 Winner											
IMDN_baseline											

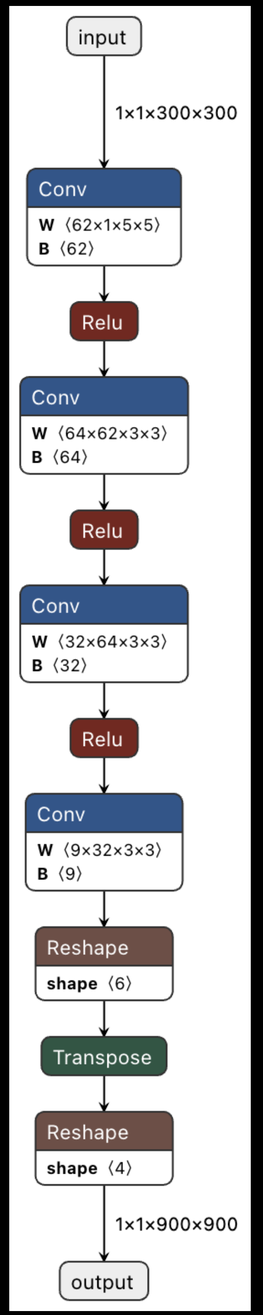
* This solution uses too much GPU memory. Image

Compatibility Matrix of SOTA SR Models

SOTA Model	XGen Baseline Compatibility	NNI Pruning Compatibility	Baseline Inference on V100 (ms)	Baseline Inference on Galaxy S10e (ms)
Twitter	✅	✅	2.39	72.46
ByteDance	✅	❌	20.97	166.00
WDSR (XGen Demo)	✅	❌	54.96	424.70
RDN	❌	❌	67.84	❌
IMDN	❌	❌	27.17	❌
Real-ERSGAN	❌	❌	200.12	❌

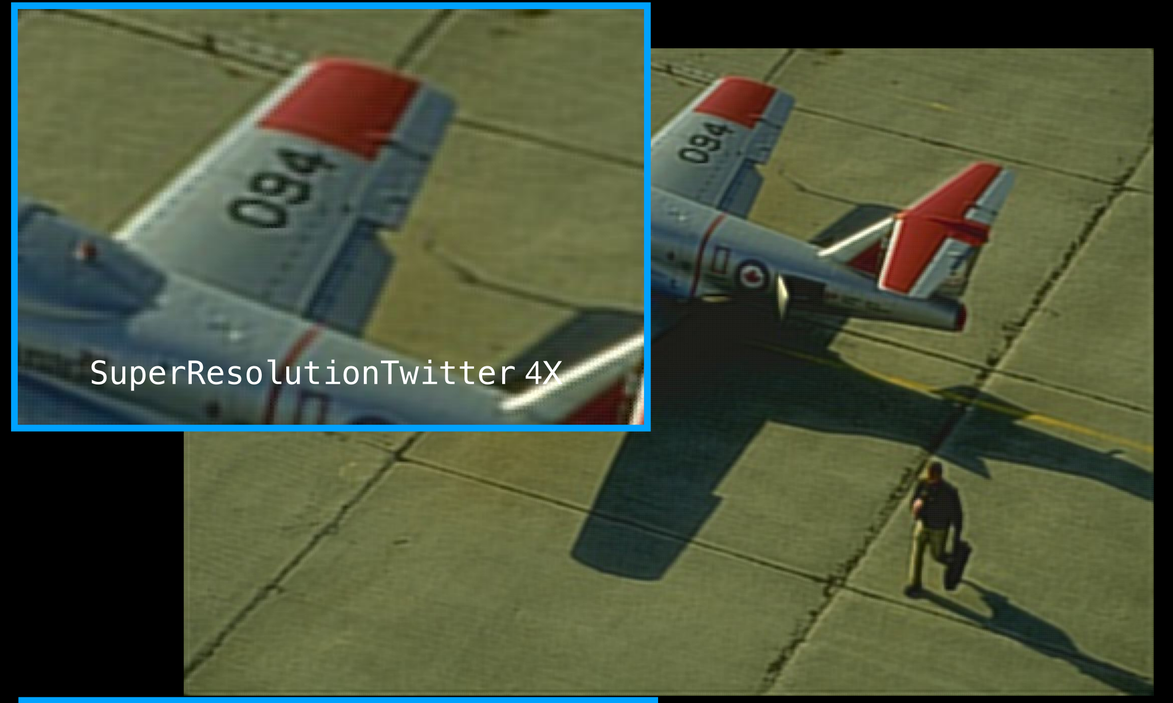
SuperResolutionTwitter

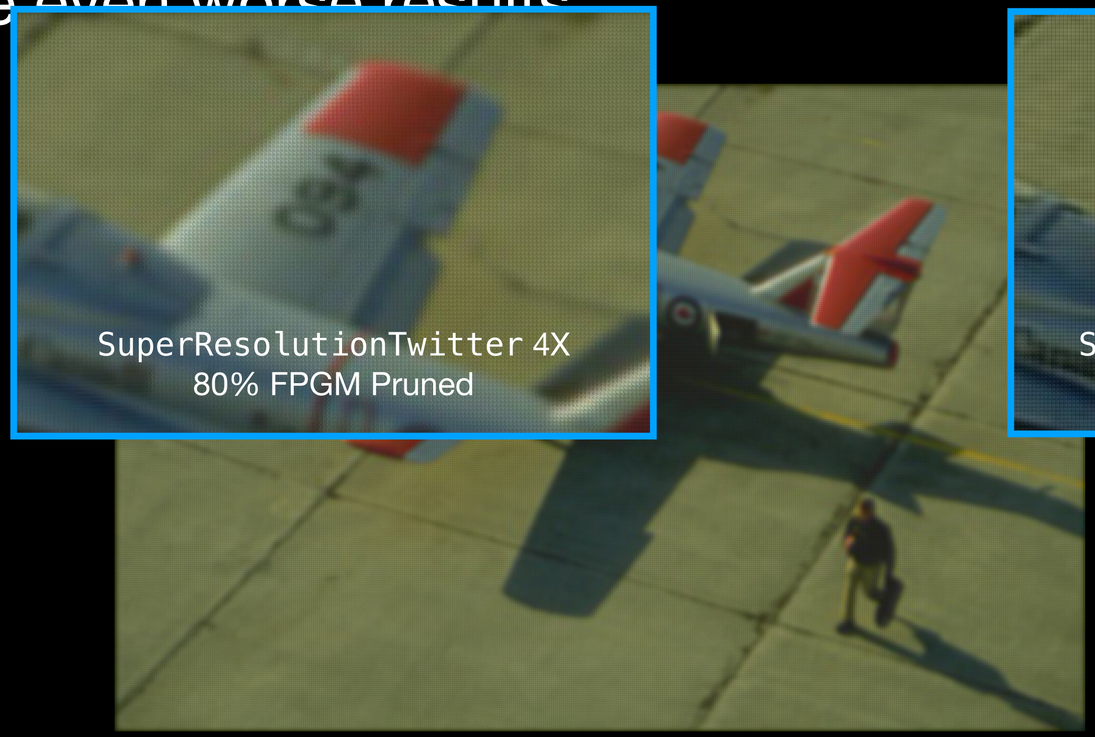
- ***Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network***, Twitter
- CVPR 2016
- https://arxiv.org/abs/1609.05158
- Trained on BSD300 dataset

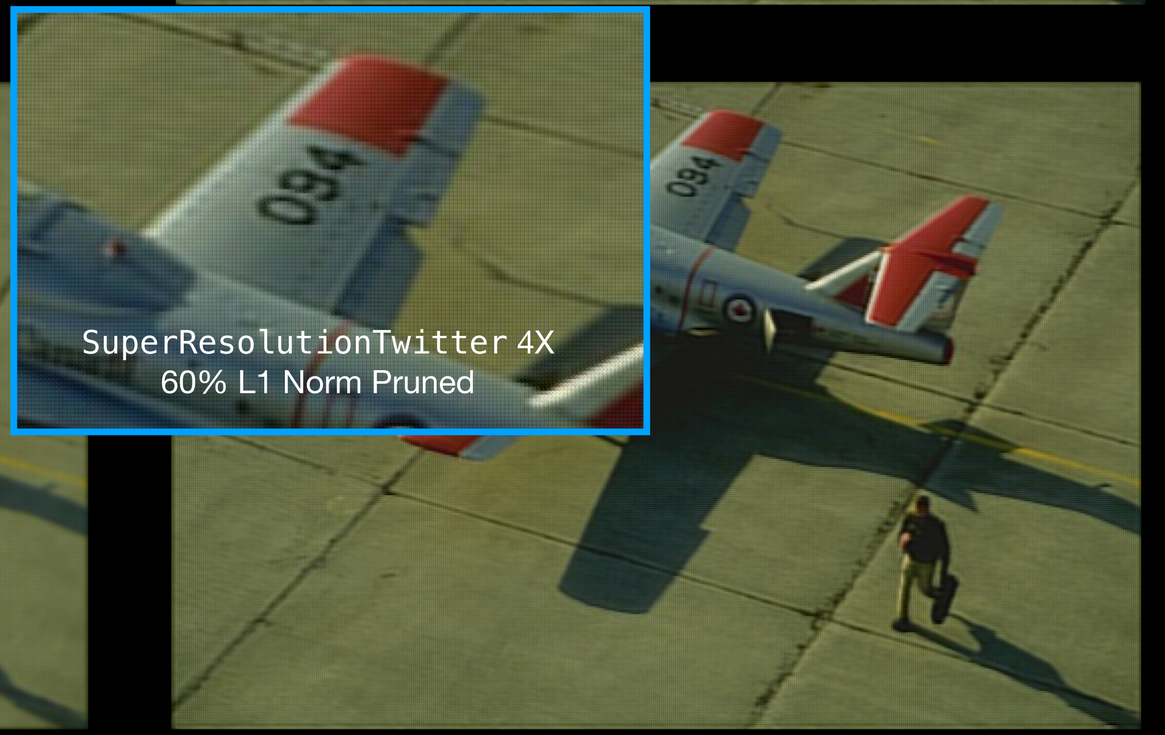


Optimizations
Pruning

- Pruning degrades quality of model
- Pruning aggressively over 80% will produce even worse results







Neural Network Intelligence

Optimizations
Quantization

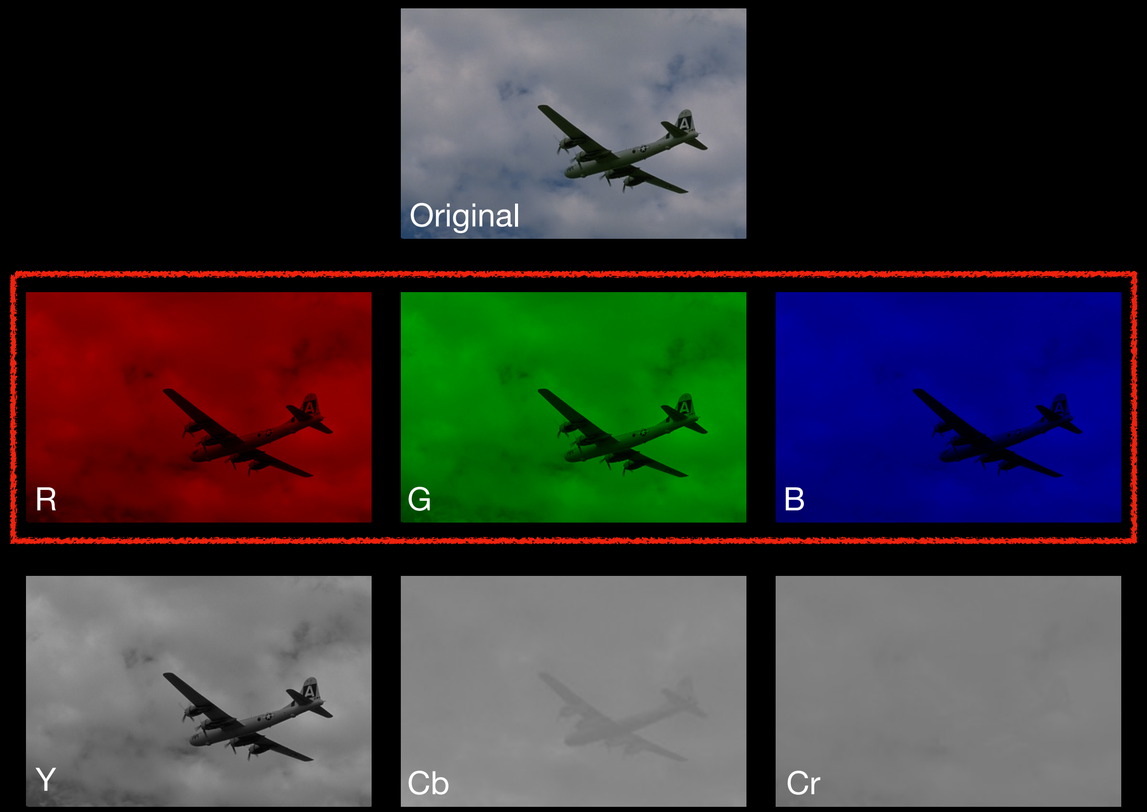
- Tried modifying the ONNX file to FP16 weights
- FP16 supported by both mobile and cloud hardware
- XGen does this for us, so no manual optimizations required

CoCoPIE

Optimizations

Color Optimization

- Used YCbCr instead of RGB channels
- Training and inference on only the luma (Y) channel
- Reduces tensor size by 3X from (N, 3, H, W) to (N, 1, H, W)
- Overhead of deconstruction and reconstruction at every frame

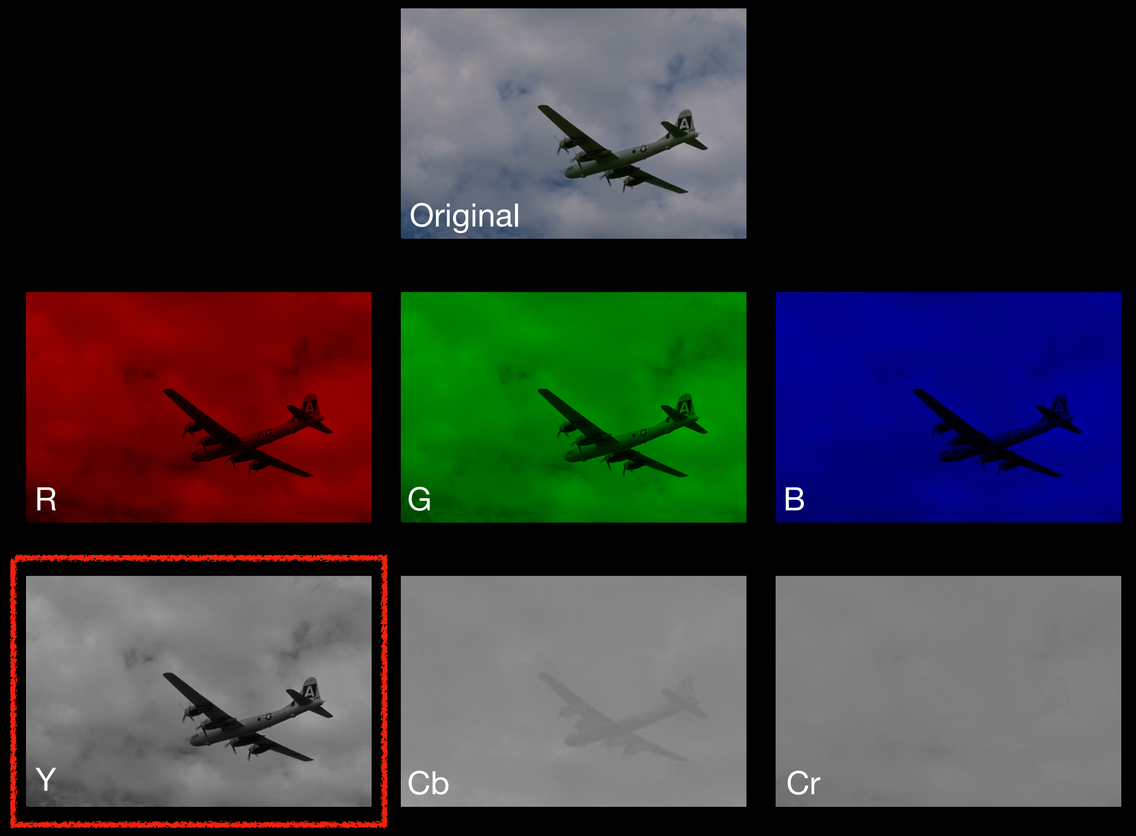


Inputs to Super Resolution Model

Optimizations

Color Optimization

- Used YCbCr instead of RGB channels
- Training and inference on only the luma (Y) channel
- Reduces tensor size by 3X from (N, 3, H, W) to (N, 1, H, W)
- Overhead of deconstruction and reconstruction at every frame

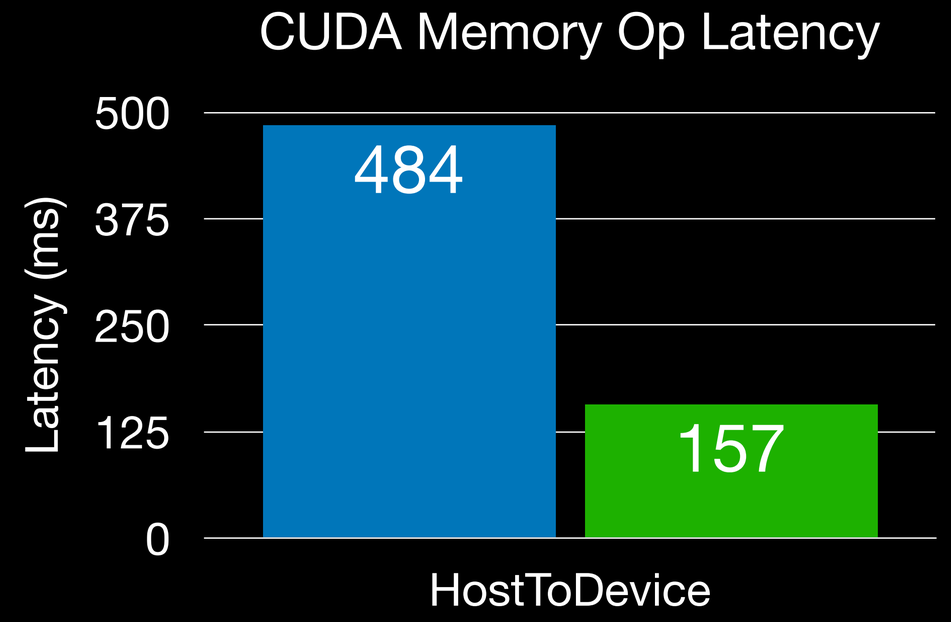


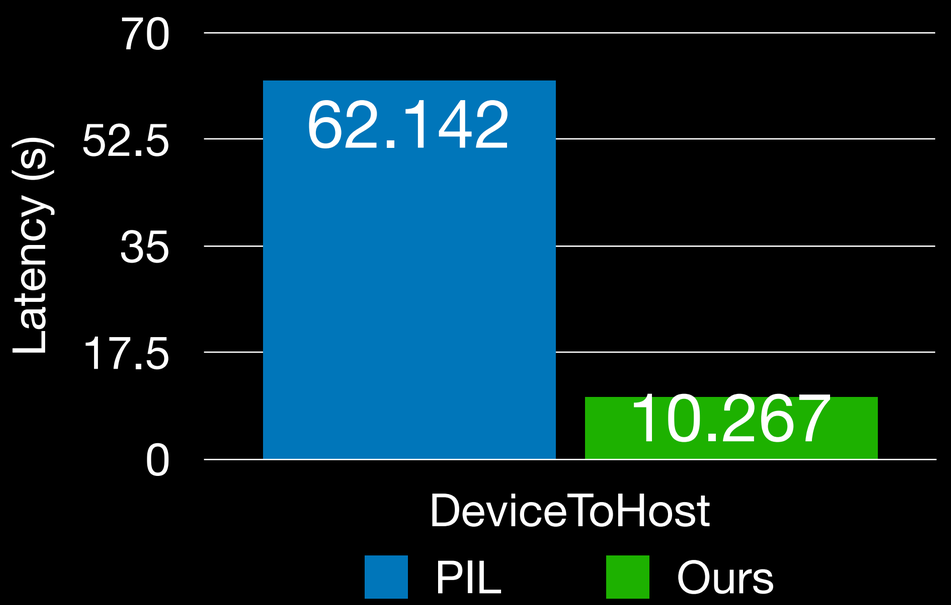
Inputs to Super Resolution Model

Optimizations

Color Optimization

- Originally used Python Imaging Library (PIL) to do the color conversion
  - PIL is written only for the CPU
- Overhead of GPU/CPU memory transfers (`cudaMemcpyDeviceToHost`/ `cudaMemcpyHostToDevice`)
- Optimized code by converting CPU operations into CUDA operations via PyTorch





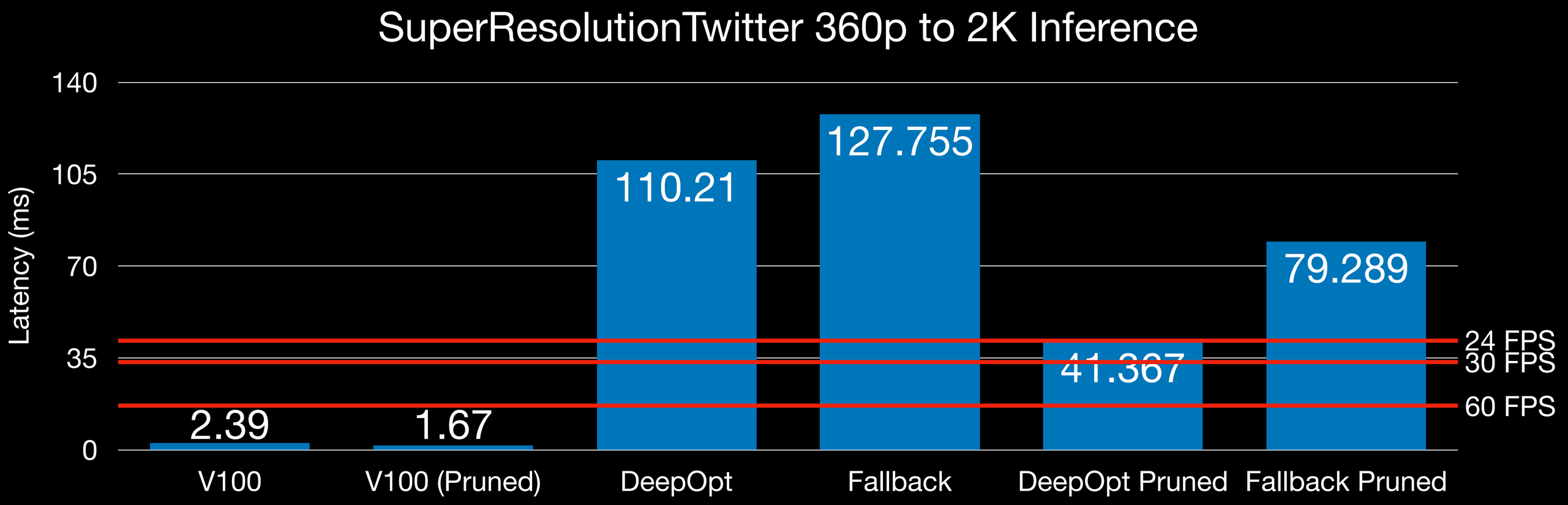
Experimental Results

Benchmark Configuration

- Benchmarked video frames upscaled from 360p to 2K resolution
  - Upscale factor set to 4X
  - From input of 640 x 360 to output 2560 x 1440

Experimental Results

Latency



Original: 244KB

Pruned: 100KB

*L1NormPruner 60% Pruned

Experimental Results

Quality

PSNR: 6.47

PSNR: 20.69

Experimental Results

Other Frameworks

- Successfully tried compilation on other frameworks
  - TVM (Apache)
  - TensorRT (NVIDIA)
  - ONNX Runtime (Microsoft)
  - CoreML (Apple)

Challenges and Lessons Learned

- Hard to achieve a workable model on XGen due to limited functionality of opset
- Limited XGen resources with GPU and smartphone
- Easy to achieve and collect results on HPC GPU, but not for mobile
- Learned more about computer vision and super resolution

Next Steps

- Try more Super Resolution architectures
  - RNN could be used for video super resolution, as video frames hold redundant data between frames
- Aggressively apply other optimizations without impacting quality
- Try different datasets

Thank You

- ARC (NCSU Systems Lab)
- Bridges-2 (PSC)
- Lab XGen Server and Samsung Galaxy S10e (Dr. Shen)

NC STATE UNIVERSITY