DES2/AIS4-3
*Invited*

# Multistage Video Up-Scaling Technique for 8K High Quality Display

**Sewhan Na[1], Unki Park[1], Hyun-Wook Lim[1], Jaeyoul Lee[1]**

sewhan.na@samsung.com
[1]System LSI Division, Samsung Electronics Co. Ltd., Hwaseong, Korea

Keywords: multistage upscale, super resolution, convolutional neural networks, jagging suppression, detail enhancement

## ABSTRACT

*A multistage video up-scaling in this paper, consisting of a linear filter, convolutional neural networks, and a hand-made adaptive scaler trained by machine learning theory, achieves a high quality 8K resolution, which is hard to obtain with a single up-scaling at low cost, in DTV application. The proposed has image restoration performance of PSNR 27.5dB in a 10-bit per color component quantized 3x scaling without jagging artifacts. A real time multistage scaler for different combinations of resolutions, compared to a single scaler at the similar quality, is implemented at one-third hardware cost.*

## 1 INTRODUCTION

Recent popularization of 8K UHD TV has led to high-resolution widespread but still hard to find real 8K (7680 × 4320 pixels) video contents. Therefore, the video up-scaling will be the main technology for a while in the 8K high quality display.

The advent of AI technology, far away from conventional interpolation methods, has given the potential for high quality up-scaling to 8K resolution. Related researches known as the super-resolution are being studied a lot, and among them, deep learning based algorithms using convolutional neural networks (CNNs) show very good performance [1]. However, there are many hurdles in implementing CNN layers as a high-resolution video processing chip that requires real-time operation over dozens to hundreds frame per second (FPS). Deep depth of CNN layers in a general neural network processor of several tera operations per second (TOPS) is very inefficient to process 8K video over 60FPS. Moreover, supporting multiple scaling ratios according to source video resolutions require a complicated network architecture and its training costs are very high. For these reasons, the dedicated hardware for 8K up-scaling algorithm is implemented in the image pipeline of a video processing data path.

In this paper, we propose a multistage video up-scaling scheme to apply a high performed CNNs based up-scaler and its peripheral functions cost-efficiently in the image pipeline. Since the hardware takes a lot of logic and memory, the processing data bit precision and depth of CNN layers are optimized to save chip area and power consumption while maintaining the image quality benefits of jaggy-less and thin edge restoration.

## 2 PROPOSED METHODS

### 2.1 Overall Architecture

An overall block diagram is shown in Figure 1. The pipeline is comprised of poly-phase linear filter, super-resolution and custom adaptive scaler.

At first, any low-resolution input is converted to a 2K fixed resolution by poly-phase filter prior to each single ratio of 2x conversion in neural networks and custom scaler. The decimal scaling ratio in the range between 1.0x and 2.0x is used in accordance with the input resolution. Secondly, a super-resolution, based on the deep neural networks, converts video frames from 2K to 4K resolution. The network model is quantized and trained by several types of image datasets for 2x fixed rate conversion. Finally, custom adaptive scaler with classified kernels are used for 4K to 8K up-conversion requiring four times higher pixel per cycle processing. The kernels are trained to maintain the jaggy-less and thin edge image features from the previous stage at relatively low hardware cost.

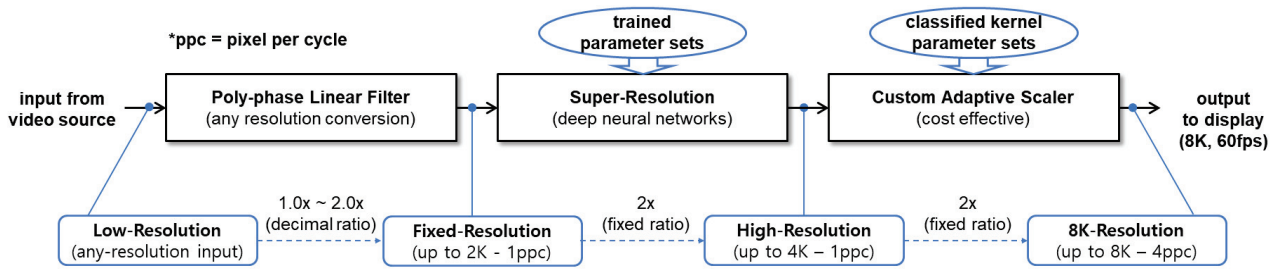



**Fig. 1 Overall Block Diagram of Resolution Conversion Steps**

### 2.2 Poly-phase Linear Filter

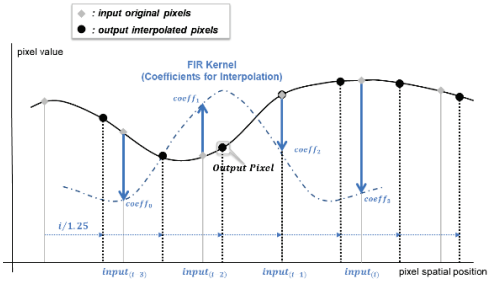
Converting from low-resolution to high-resolution is to increase the sample of pixels called interpolation, and there are well-known nearest, bilinear and bicubic methods. These are simple but have quality issues, when used for 8K high-resolution restoration, such as jaggies on the edges and image blur caused by high frequency component loss. However, these interpolations have the advantage of usage when converting any-resolution to fixed resolution.

To design an any-resolution up-convertible interpolator with easy controllability of the image quality factors, we propose a fine tunable finite impulse response (FIR) filter. The Equation 1 for resampling has adjusting parameters, which have trade-off relations between sharpness and aliasing artifact, by a sinc function modified with Gaussian distribution deviated by $\sigma$. The transfer function is windowed by a Kaiser window [2], where $ctrl_{sharpness}$ is a sharpness control parameter, $ctrl_{aliasing}$ is a aliasing control parameter and $i$ is a pixel position.

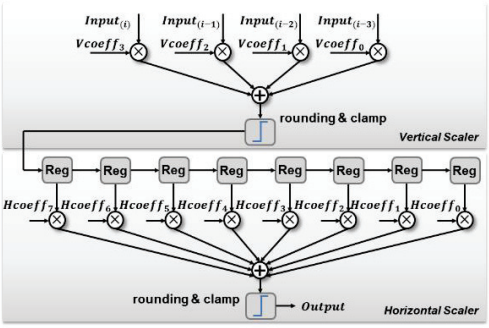
$$h[i] = \left\{\frac{\sin(x)}{x} - ctrl_{sharpness} \cdot \exp\left(-\frac{x^2}{2\sigma}\right)\right\} \times Kaiser(i),$$

$$x = i \times \pi \times ctrl_{aliasing} \qquad (1)$$

The $h[i]$ is truncated by 4-taps windowing kernels for vertical and 8-taps for horizontal domains to be implemented by a low cost poly-phase linear filter as shown in Figure 2. To reduce the line buffers for processing vertical resolution, the number of taps are smaller than that of the horizontal direction.




(a) Interpolation with kernel (example for 1.25x, 4-taps)




(b) Poly-phase filtering hardware structure

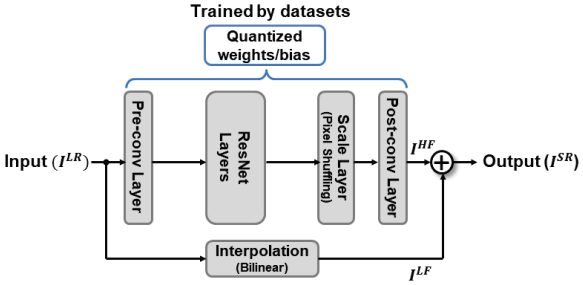
**Fig. 2 First Step by Poly-Phase Linear Filter**

### 2.3 Super-Resolution (Deep Neural Networks)

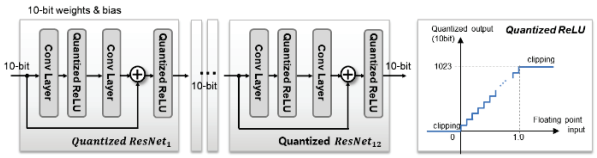
The middle step shown in Figure 3 is up-scaling by trained CNNs to make the high quality restoration. The main component of the model is a residual neural networks (ResNet [3]) with pre and post-convolutional layers. A pixel shuffling layer converts the resolution of input ($I^{LR}$), passing ResNet layers, to target resolution and then the output of post-convolutional layer ($I^{HF}$) is added to bilinear scaled input ($I^{LF}$) to restore the final up-converted output($I^{SR}$) as Equation 2.

$$I^{SR} = I^{HF} + I^{LF} = CNNs(I^{LR}) + Bilinear(I^{LR}) \qquad (2)$$

We stacked 10-bit quantized 12 ReNet blocks, each one layer for the pre and post-convolutional block, and one layer for the pixel shuffling in order to be a reasonable hardware size.




(a) Deep neural networks model structure




(b) Quantized ResNet block

**Fig. 3 Middle (Main) Step by Super-Resolution**

### 2.4 Custom Adaptive Scaler

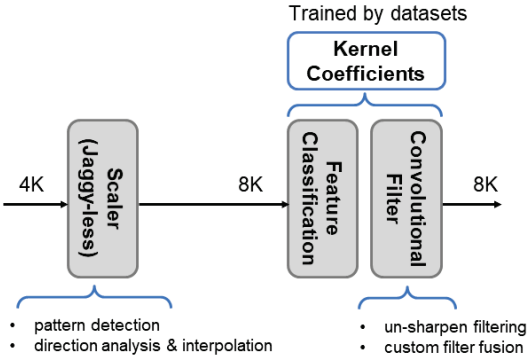
We design the final step with a custom adaptive scaler, shown in Figure 4, composed of a jaggy-less scaler and adaptive convolutional filter. The scaler converts fixed resolution of 4K input to 8K output to design the model simple and cost-effective because this step has to address 8K resolution, which requires four times as much hardware as 4K in the same algorithm. To restore pixels without jaggy, 13 levels of categorization for edge detection is considered.

The next custom adaptive filter is an unsharp mask fused with kernels learned by least-square method in Equation 3 with m data samples in a certain class. Pixel $\{y'_k\}$ is generated by convolution of pixels from degraded input $\{x_{k0}, x_{k1}, \ldots, x_{k12}\}$ and kernel coefficients $\{w_{k0}, w_{k1}, \ldots, w_{k13}\}$ where k is a certain class number from the feature classification block. Least-square method by off-line software iteration is used to solve

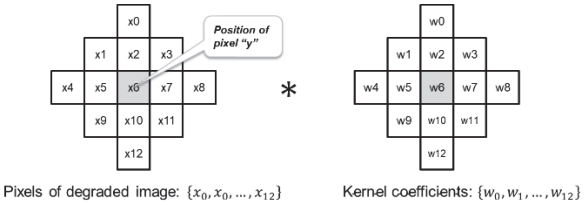
$e^2 = argmin(\sum_{k=0}^{m} e_k^2)$ and find the coefficients in each class, where $\{y_k\}$ is a target pixel without degradation from prepared datasets.

$$e_k = y_k - y'_k = y_k - \sum_{i=0}^{m} x_{ki} \cdot w_{ki} \quad (3)$$

The kernel coefficients for the convolutional filter are trained by k = 31 classified image features and used to enhance the details after up-scaling process of previous block.




(a) Block diagram of custom adaptive scaler




(b) 13-taps of custom filter kernel for convolution

**Fig. 4 Last Step by Custom Adaptive Scaler**

## 3 EXPERIMENTS

The proposed video up-scaling pipeline is evaluated by a software modeling before it is implemented in a hardware. Summary of proposed algorithms and hardware specifications are in Table 1.

The deep neural networks and custom adaptive filters are trained each by DIV2K [4] and custom dataset images. The measured PSNR on the DIV2K validation set is 30.6dB, 27.5dB, and 26.1dB in the case of 2x, 3x, and 4x scaling with 10-bit quantized parameters and processing resolution.

Figure 5 shows the simulation results from each output stage in the case of total 6x scaling ratio and a comparison of the results from bicubic, proposed (super-resolution and custom adaptive scaler), and two cascaded super-resolution blocks to make 4x scaling ratio. The proposed pipeline results in (a) show the superior performance compared to conventional bicubic results especially the quality of jaggy in the edge and thin edge restoration. Comparing to the architecture, the results in (b) show that the final output image of proposed architecture using one super-resolution and then one custom adaptive filter, which is one-third of hardware cost, is similar to that of cascaded two super-resolution scaling blocks without image quality degradation.

**Table 1 Summary of Specifications**

| Algorithm / Architecture | | Input Resolution | Output Resolution | Hardware Costs |
|---|---|---|---|---|
| Multistage video up-Scaling | • YUV444 / 60fps video<br>• 10-bit per color component & processing parameter | 960x540 ~ 7680x4320 | 7680x4320 (1.0x ~8.0x) | • # of logic:<br>- mult.: 7.2K<br>- adder: 5.6K<br>• memory<br>- 13.2Mbits |
| [Step1] Poly-phase Linear Filter | • filter taps/phase:<br>- horizontal 8-taps / 64phases<br>- vertical 4-taps / 32phases<br>• order of processing:<br>- vertical→horizontal domain<br>• FIR design:<br>- kaiser window kernel | 960x540 ~ 1920x1080 | 1920x1080 (1.0x ~ 2.0x) | • # of logic:<br>- mult.: 12<br>- adder: 2<br>• memory:<br>- 0.1Kbits |
| [Step2] Super-Resolution (CNNs) | • neural networks:<br>- CNNs, 27 layers, 3x3 kernel<br>• # of total parameters: 9.6K<br>• including pixel shuffling & bilinear<br>• skip connection and addition | 1920x1080 | 3840x2160 (2x) | • # of logic:<br>- mult.: 6K<br>- adder: 5K<br>• memory:<br>- 12Mbits |
| [Step3] Custom Adaptive Scaler | • jaggy-less scaling:<br>- 13 categorization of edge pattern<br>• adaptive filter:<br>- 31 classified kernels (learned) | 3840x2160 | 7680x4320 (2x) | • # of logic:<br>- mult.: 1.2K<br>- adder: 0.6K<br>• memory:<br>- 1.2Mbits |

## 4 CONCLUSION

In the paper, we proposed a video up-scaler for 8K display through a multistage pipeline. The pipeline is divided into three different stages of methods to optimally implement their features and benefits in a chip, which are any-resolution source input conversion, detail preserved high quality image restoration without artifacts, and real-time video processing. The up-conversion from 2K to 4K resolution, which determines most of the final output image quality, is performed at the intermediate stage of trained deep neural networks. Experiments show that the image quality of proposed scheme is similar to that of a single stage up-scaler which is three times more expensive. The proposed is applicable to 8K display driving products such as timing controller (T-CON) and DTV-SoC.

## REFERENCES


[1] Bee Lim et al., "Enhanced Deep Residual Networks for Single Image Super-Resolution," CVPR Workshops (2017)

[2] Oppenheim et al., "Discrete-time Signal Processing," 2nd ed., Prentice Hall (1999)

[3] Kaiming He et al., "Deep Residual Learning for Image Recognition," CVPR (2016)

[4] R. Timofte et al., "Ntire 2017 challenge on single image super-resolution: Methods and results," CVPR Workshops (2017)

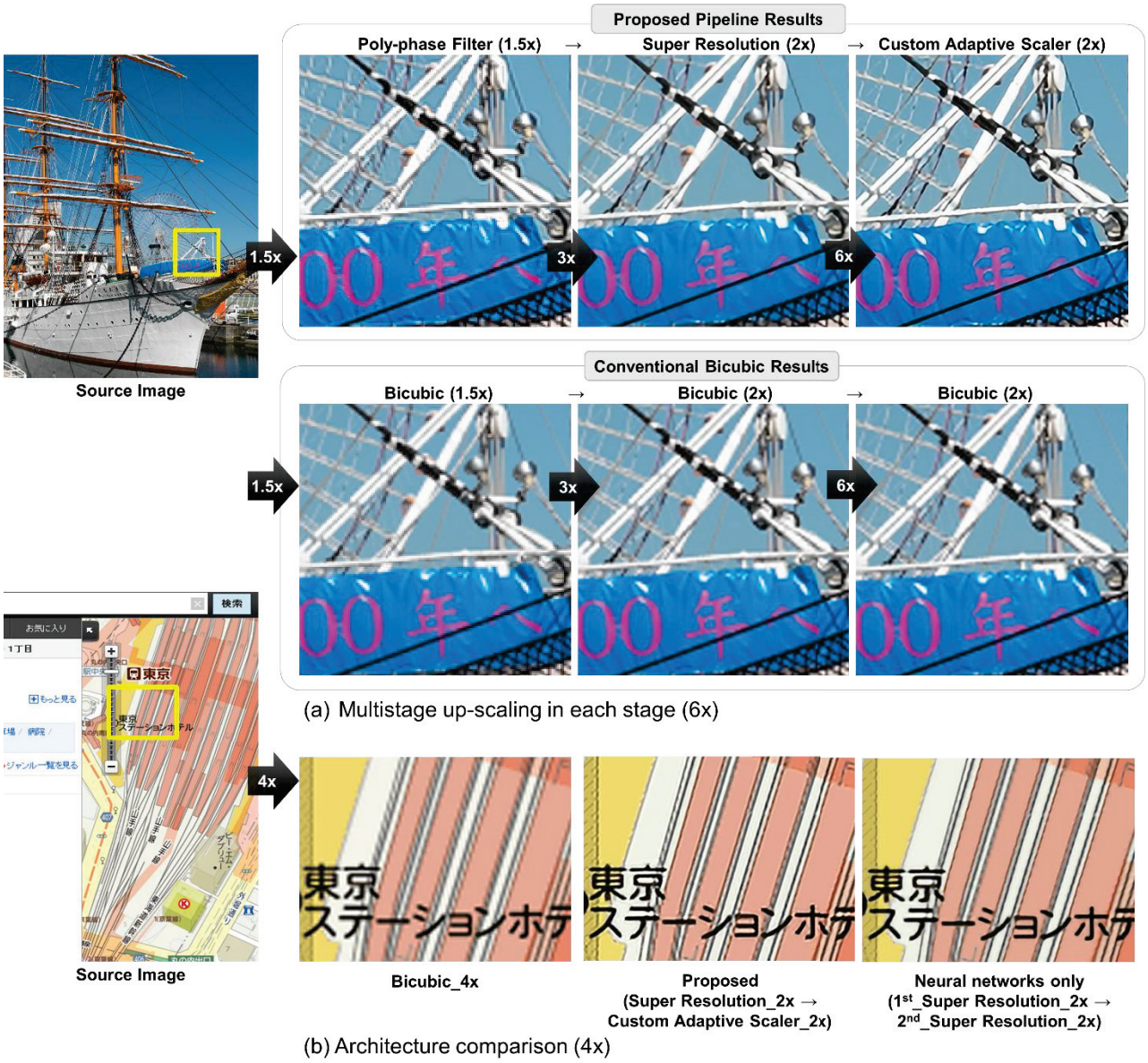



**Fig. 5 Evaluation Results.** (a) Each image from the proposed up-scaling pipeline comparing with conventional bicubic. (b) Comparison of jaggy-less and thin edge restoration features in 4x conversion