KSII The 8th International Conference on Internet (ICONI) 2016 Symposium.

A New Frame Rate Up Conversion Quality Enhancement Method using Deep Convolutional Neural Network and Temporal Difference Map

Sangchul Kim, Seungbin Lee, Kwangsoo Shin and Jongho Nang
Department of Computer Science and Engineering, Sogang University
Seoul, South Korea
[e-mail: sckim@sogang.ac.kr, mercileesb@sogang.ac.kr, kepler92@sogang.ac.kr and jhnang@sogang.ac.kr]
*Corresponding author: Sangchul Kim

Abstract

This paper presents post-processing algorithm to enhance the quality of FRUC. Our proposed method uses denoise filter trained by a deep convolutional network to reduce the noise of interpolated frames and then apply sharpening process considering temporal redundancy. Frame Rate Up Conversion (FRUC) is a method that inserts interpolated frames between source frames for smoother video playback. Interpolated frames from conventional FRUC have quality issues caused by incorrectly estimated motion vectors, hole effect, and object overlapping. To address these issues, we propose a method that combines a deep convolutional neural network (CNN)-based denoise filter and a foreground/background classifier that strengthens outer edges of objects. In our experiment, the proposed method improved the quality of frames interpolated through FRUC by between 1.52dB to 2.13dB.

Keywords: frame rate up conversion, FRUC, deep learning, deep convolutional neural network, CNN, image restoration, image reconstruction, video processing, video enhancement

1. Introduction

Frame Rate Up Conversion (FRUC) is a method that inserts interpolated frames between video frames for smoother video playback. FRUC is used for improving video compression ratio by selectively skipping frames [1] and for fitting 24fps/30fps videos to a 60fps display device for smoother playback [2]. The block-based motion-compensated FRUC, which is the current state-of-the-art method, places objects on estimated trajectory of each object. In that method, motion vectors are at times incorrectly estimated and they result in empty regions where pixels on screen are not mapped at all (hole effect) or mapped pixels overlap (ghosting effect).Previous work[3, 4] proposed to enhance motion vector estimation, but the underlying

This research was supported by the MISP(Ministry of Science, ICT & Future Planning), Korea, under the National Program for SW in Excellence)(R2215-16-1003) supervised by the IITP(Institute for Information & communications Technology Promotion), and by the ICT R&D program of MSIP/IITP. [R0126-16-1112, Development of Media Application Framework based on Multi-modality which enables Personal Media Reconstruction]

principle, SAD (Sum of Absolute Difference), which is used for block-based motion estimation, has been found not very reliable for identifying accurate motion vectors at all times. Existing FRUC methods[5-7] interpolate frames in blending mode, which unfortunately also blend background and objects together, produce ghosting effects. Our previous work[2] proposed a path-based frame interpolation method, but the quality of the interpolated frames were influenced by the accuracy of motion vectors and size of hole area. In [8, 9], hole effect was addressed with spatial interpolation by exploiting neighboring pixels around holes, but frames become distorted when the size of a hole is significantly large. In this paper, our approach to the issues described above is as follows: frames are interpolated via unidirectional FRUC, hole areas are treated with spatial interpolation, and the noise generated after spatial interpolation is reduced with a deep convolutional neural network-based denoise filter. Blurring effect by the denoise filter is alleviated by strengthening outer edges between foreground and background as identified by means of temporal redundancy. In our experiment, image quality was improved by between 1.52dB and 2.13dB. This paper is organized as follows. Section 2 discusses related work. Section 3 describes our proposed method, followed by experiment and analysis in Section 4. This paper is concluded in Section 5.

2. Related works

2.1 Frame Rate Up Conversion

Frame Rate Up Conversion (FRUC) is a method that inserts interpolated frames between source video frames for smooth video playback. Depending on how motions are estimated, FRUC can be categorized as unidirectional and bidirectional.

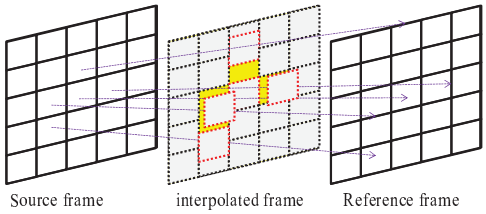
Fig. 1 illustrates unidirectional and bidirectional FRUC. In each illustration, object motions are estimated for each block.

In unidirectional FRUC, blocks are identified that have the smallest sum of absolute difference (SAD) between the source frame and the referenced frame. Incorrectly estimated motion vectors can produce the hole effect, where regions in the interpolated frame are not mapped with pixel, or the ghosting effect, where regions are mapped with pixels more than once.

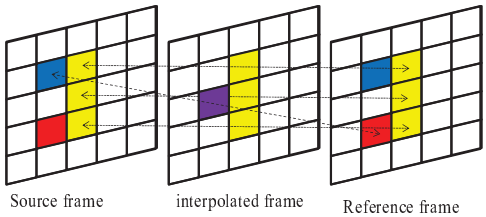
In some previous work[8, 9], areas affected by hole effect was treated with spatial interpolation, but filling in significantly large hole regions produced awkward-looking results.

To reduce hole effects, bidirectional FRUC[7] was proposed, in which motion is estimated in two directions for interpolated frames instead of in one direction. It had a shortcoming of producing rough pixel mappings at block edges.

Hence, pixels were overlapped to make edges smooth, but it had the side effect of introducing blurring effect. Ghosting and overlapping induced by blending-based interpolation were addressed in a previous work[2] by pixel-mapping that exploited temporal redundancy, but it did not completely remove hole effect. The root cause of hole effect, ghosting effect, and overlapping is incorrect estimation of block motions.



(a) unidirectional FRUC, yellow region of interpolated frame is hole area



(b) bidirectional FRUC, purple block of interpolated frame is result caused by inaccurate motion estimation

Fig. 1. An example of Frame Rate Up Conversion as Illustrated in [2]

2.2 Convolutional Neural Networks

Neural network algorithms mimic how the human brain perceives objects internally, so they simulate neurons and synapses with networks of graphs and nodes. The neural network algorithm was first proposed to solve problems that are hard to approach intuitively, but optimizing the tens of thousands of interconnected nodes was thought to be intractable.

Hinton et al. proved this optimization problem to be tractable by utilizing pre-processing network that consists of restricted Boltzmann Machines (RBM) in [10]; Lecun et al,

implemented a handwriting recognition system based on deep convolutional neural networks (CNN)[11], which is a type of neural network with multiple hidden layers considering spatial redundancy. CNN is popular in the domain of visual recognition.

Training a neural network with a large volume of samples requires a significant processing capacity, but this is now handled with parallel processing on general-purpose GPU (GPGPU). CNN has been used in recent work of image restoration [12][13] as well.

2.3 Deep learning for Image Restoration

With CNN based on multi-layer perception (MLP), images can be denoised and restored[14]. The net result of fully connected neural network is inferior to the result of CNN. Denoising filter was trained with CNN by Eigen et al.[12] and the result was promising.

However, this approach required much processing capacity and it used individually trained filters such as 'rainy' or 'dirty', making their method inadequate for general cases. Super resolution based on end-to-end mapping[13] requires far less processing capacity while producing great results, but it is not suitable for denoising because this is essentially an edge-sharpening method.

3. Post-Processing for FRUC Quality Enhancement

3.1. Convolutional Network Structure

A popular strategy in image restoration is to densely extract patches and then represent them by a set of pre-trained filters[13]. Our proposed method is inspired by Dong et al.[13] because end-to-end mapping and shallow configuration allows for fast processing.

Increased number of convolutional layers and feature maps increases video quality but it also increases computational complexity.

In our work for this paper, 4 convolutional layers and 4 hyper parameters were used with 32 out nodes for each layer as illustrated in **Fig. 2**, and parameters were set at $F_1 = 7$, $F_2 = 3$, $F_3 = 1$, $F_4 = 5$ With $F_3 = 1$, non-linearity was increased in the convolutional layers, which were configured to correspond with fully connected layers. To make end-to-end mapping less linear, we also used a 1x1 filter in the third convolution layer.

3.2. Image Restoration

As C.Dong et al[13]. discussed in end-to-end mapping-based filter generation and restoration, image restoration consists of **i**. patch extraction and representation, and **ii**. non-linear mapping and **iii**. restoration stage. This section describes mapping functions of each stage.

Equation (1) maps the first convolution weight matrix to the patch while Equation (2) is a non-linear mapping to next layer. The purpose is to make the regions surrounding noisy area more densely convoluted.

In [13], the rectified linear unit (ReLU, max(0, x)) is applied to the mapping equation in each layer, however, our proposed method chose not to so that the weights in Equations (1) and (2) can be preserved.

Equation (3) is a process that determines the value of each pixel, for filtering in the surrounding area with Equation (4), so the pixel value has to be applied by ReLU. Hence, application of ReLU is justified. Equation (4) inserts 0 to 255 pixel values for padding to prevent both pixel value underflow and overflow.

In our experiment, resulted in better PSNR when applied RELU selectively than non-selectively. After all of the pixels and their surrounding area have been filtered, the output image can be obtained.

i. Patch extraction and representation

$$F_1(Y) = W_1 * Y + B_1 \quad (1)$$

ii. Non-linear mapping

$$F_2(Y) = W_2 * F_1(Y) + B_2 \quad (2)$$

$$F_3(Y) = \max(W_3 * F_2(Y) + B_3, 0) \quad (3)$$

iii. Restoration (reconstruction)

$$F(Y) = \min(\max(W_4 * F_3(Y) + B_4), 0), 255) \quad (4)$$

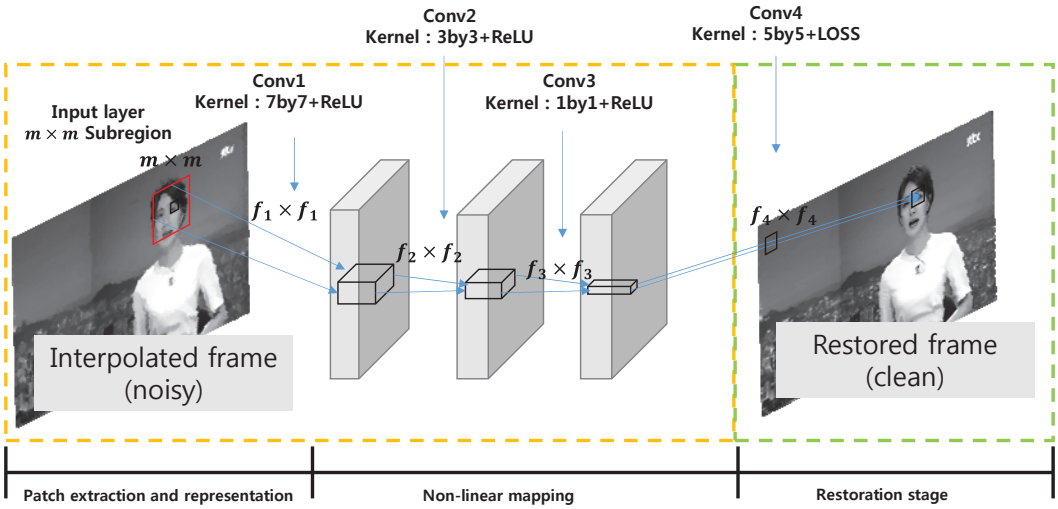


Fig. 2. Proposed convolutional neural network structure for reduction noise of interpolated frames

3.3. Edge Sharpening by Classifying Fore/Background based on Temporal Difference Map

Hole effect could arise in a unidirectional FRUC when motion vectors are incorrectly estimated and as a result objects moving along the motion vector are placed in the wrong place.

Whether a hole region is in the background or on top of an object could be determined in the frames following the source frame or preceding the reference frame.

Hole effect appears around object edges because there is no matching vector for background after an object is relocated. In the edge regions, foreground and background can be distinguished. Video quality can be improved if edge regions can be clearly mapped.

Equation (5) finds the difference frame between the reference frame and the altered source frame, which has been shifted along the trajectory specified by the global motion vector. A low pixel value in the difference map indicates object motion is small. An object with small values in the difference frame can be regarded as a fixed object or the background. Therefore, the pixel table is mapped to Obj_{same}.

When the difference is large, the previous object has been relocated and a new object has been placed in the pixel coordinates. Equation (7) is used to determine whether the object comes from the next frame or the previous frame as the foreground/background pixel mapping applying equation (8), which is fill background pixels from whether frames next or previous frame except for same case. At the same case, the equation (8) fill the pixel as averaging pixel between the two frames.

3.4. Training

In the proposed architecture, an end-to-end mapping function is trained for estimating the parameters $\Theta = \{W_1, W_2, W_3, W_4, B_1, B_2, B_3, B_4\}$ of the convolutional network.

PSNR (peak signal-to-noise ratio) is a common metric for measuring image quality deterioration and preservation.

The lower the MSE (mean squared error) is for an image, the higher the PSNR is, hence we set the objective function to find the optimal parameter as in Equation (9). F(Y;m) is the restored output frame from the interpolated frame Yby using the filter generated through m, and X is the groundtruth corresponding to the interpolated frame.

$$D(dx, dy, I_t) = \frac{d_{I_t}}{dt}(dx, dy) \tag{5}$$

$$\begin{aligned} &if\ D(dx, dy, I_t) < \varepsilon_o \rightarrow T(x, y) \\ &= Obj_{same} \\ &else\ then \rightarrow T(x, y) \\ &= Obj_{other} \end{aligned} \tag{6}$$

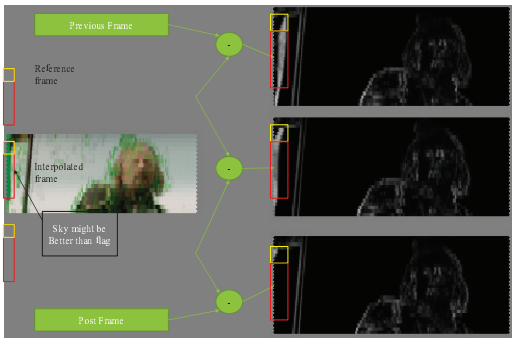


Fig. 3. (a) Classifying foreground versus background in a hole area, (b) difference map example of adjacent frames from [2]

$$\begin{cases} if\,(D(dx_{t-1}, dy_{t-1}, I_{t-1}) > \varepsilon_o \;\&\; D(dx_{t+1}, dy_{t+1}, I_{t+1}) > \varepsilon_o) \\ \quad \rightarrow \dot{T}(x,y) = Obj_o \\ if\,(D(dx_{t-1}, dy_{t-1}, I_{t-1}) < \varepsilon_o \;\&\; D(dx_{t+1}, dy_{t+1}, I_{t+1}) > \varepsilon_o) \\ \quad \rightarrow \dot{T}(x,y) = Obj_o \\ if\,(D(dx_{t-1}, dy_{t-1}, I_{t-1}) > \varepsilon_o \;\&\; D(dx_{t+1}, dy_{t+1}, I_{t+1}) < \varepsilon_o) \\ \quad \rightarrow \dot{T}(x,y) = Obj_o \\ if\,(D(dx_{t-1}, dy_{t-1}, I_{t-1}) < \varepsilon_o \;\&\; D(dx_{t+1}, dy_{t+1}, I_{t+1}) < \varepsilon_o) \\ \quad \rightarrow \dot{T}(x,y) = Obj_o \end{cases} \quad (7)$$

$$when\;\; T(x,y) = Obj_{other}$$

$$\begin{aligned} &if\,\dot{T}(x,y)\,is\,Obj_b \rightarrow \dot{F}(x,y) \\ &= F_{ref-1}(x,y) \\ &else\,if\,\dot{T}(x,y)\,is\,Obj_f \rightarrow \dot{F}(x,y) \\ &= F_{src+1}(x,y) \\ &else\,if\,\dot{T}(x,y)\,is\,Obj_{same} \rightarrow \dot{F}(x,y) \\ &= {F_{ref}(x,y)}/{1-t_{elapsed}} + {F_{src}(x,y)}/{t_{elapsed}} \end{aligned} \quad (8)$$

$$\dot{\Theta} = \arg\min_{\Theta_m}\left(\sum_{i=1}^{n}\left\|F(Y_i;\Theta_m) - X_i\right\|^2\right) \quad (9)$$

$$L(\Theta) = \frac{1}{n}\sum_{i=1}^{n}\left\|F(Y_i;\Theta_m) - X_i\right\|^2) \quad (10)$$

Therefore, Equation (10) was applied, which takes the final loss function of the network as MSE. To obtain densely patched extraction, the interpolated frame Yand the groundtruth frame X were cropped to the same size.

The patched images Yi were used in training the CNN and the groundtruth images Xi were used as label data for the CNN.

4. Experiments

4.1. Dataset

For training, we chose news programs and documentary videos with the resolution of 640x352 and down-sampled them by reducing the frame rate by half, (for example, original 30fps video convert to 15fps as a training data.)

The discarded frames were designated as the label set, and the frames interpolated through FRUC from the down-sampled video as the training set. Out of these frames, 100 frames were randomly selected to extract 33x33 sliding image patches.

Approximately 300,000 such patches were used for training. We randomly selected 30 frames from a Korean news program, 30 frames from a documentary film and 30 frames from another documentary film, totalling in 90 frames as a test data set.

4.2. Comparisons to existing methods

A. Existing Denoise filter vs Our CNN filter

Fig. 4 charts the average PSNR over the test set after the denoise filter has been applied to the interpolated frames.

The proposed method shows a significantly improved performance over existing denoise filters. The reason is that the proposed denoise filter has been trained for localized image patches through CNN.

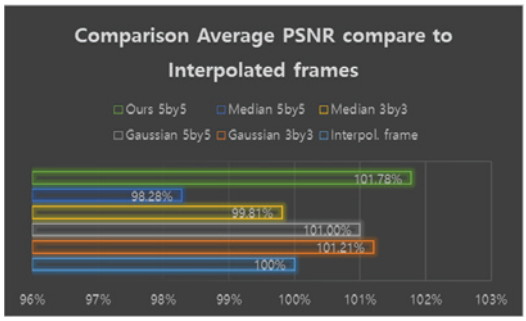


Fig. 4. Comparison of Average PSNR (in percent) for Different Filters (Applied to Interpolated Frames

Fig. 5 contrasts the output by existing filters and by our proposed denoise filter on an interpolated frame. In the images, edges are preserved well and noise is significantly reduced with our approach of using localized filters for end-to-end mapping in contrast to the Gaussian

or the median filters which blur localized regions indiscriminately.

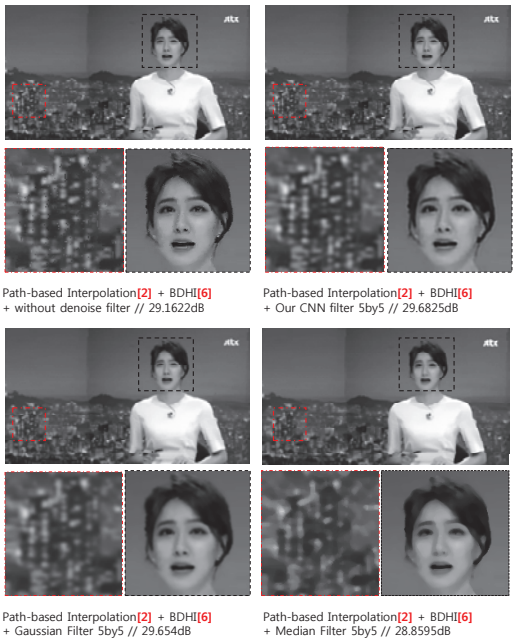


Fig. 5. Comparison of applying different denoise filters to an interpolated frame

Table 1. Comparison of PSNR for different interpolation methods

	(1) : A+B	(2) : A+B+C	(3) : A+B+D	(4) : A+B+C+**E**	(5) : A+B+**D**+**E**
Stefan	24.47dB	24.91dB	24.97dB	25.33dB	**25.99dB**
Mobile	23.10dB	23.37dB	23.47dB	23.44dB	**23.76dB**
Foreman	30.45dB	30.97dB	31.08dB	31.22dB	**32.95dB**
Flower	29.17dB	29.65dB	30.03dB	29.99dB	**30.31dB**
News+ Documentary	28.76dB	29.01dB	29.28dB	29.97dB	**30.89dB**

A : Path-based Interpolation[2], **B** : Hole Reduction [6], **C** : Gaussian Filter, **D** : CNN Filter(Ours), **E** : Adapted F/B BG Classifier(Ours)

B. Sharpening Outer Edge using Classifying Fore/Background

Fig. 6 illustrates an output example of edge sharpening with our foreground/background classifier after the image has been denoised with a our filter by CNN.

Fig. 6-1 has noise induced by motion vector incorrectly estimated during the basic FRUC as well as by the hole reduction process.

Fig. 6-2 is an output denoised with the proposed CNN filter,

whereas **Fig. 6-3** is the result of adjusting incorrectly mapped pixels in the edge regions, which sharpened outer edges, after using the adaptive foreground/background classifier.

Table 1 compares various combinations of filters and adaptive foreground/background classifier.

Columns (2) and (3) show results for frames that have been denoised; both have higher PSNRs compared to Column (1), which is the result of FRUC with no post-processing afterwards.

Column (2) is for frames that have been denoised indiscriminately while frames for Column (3) have been applied with locally optimal filters.

However, the frames for Columns (2) and (3) have lost significant amount of edge information so the image quality is worse than the frames for Columns (4) and (5), where the edges in the background and foreground have been restored.

Frames for Column (5) exhibit the best image quality; they have been treated with locally optimized denoise filters and the outer edges for foreground/background have been restored.

5. Conclusions and Future works

This paper proposed a convolutional neural network-based denoise filter to reduce noise that emerges as a result of hole-effect reduction process during frame rate up conversion. A foreground/background classifier was employed to address the blurring issue of denoise filter, and this significantly enhanced outer edge of objects.

The quality of interpolated frames was significantly improved after applying FRUC in our experiments. However, inner edges of objects still remain blurred, so this issue will be explored in our future work.

In this paper, experiments were performed only for the FRUC of our previous work [2], but other FRUC methods will be experimentally observed if our proposed method is applicable because it should work as a post-processing method for any interpolated frames regardless of the choice of FRUC.

References

[1] J. Nang, S. Kim, H. Lee, "Classifying Useful Motion Vector for Efficient Frame Rate Up Conversion of MC-DCT Encoded

Video Streams," *Journal of Information Science and Engineering*, vol. 30, no. 06, pp. 1755-1771, 2014.

[2] S. Kim, Doohee Oh, Jongho Nang, "A New Path based Interpolation using Object Motion Frame Rate Up Conversion," *in Proc. of The 5th IEEE International Conference on Consumer Electronics – Berlin(ICCE-Berlin)*, pp. 108-112, 2015.

[3] D. Yoo, S. Kang, Y. Kim, "Direction-Select Motion Estimation for Motion-Compensated Frame Rate Up Conversion," *Journal of Display Technology*, vol. 09, no. 10, pp. 840-850, 2013.

[4] D. Kim, H. Lim, H. Park, "Iterative True Motion Estimation for Motion-Compensated Frame Interpolation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.23, no.03, pp.445-454, 2013.

[5] S. Kang, S. Yoo and Y. Kim, "Dual Motion Estimation for Frame Rate Up Conversion," *IEEE Transactions on Circuits System Video Technology*, vol.20, no.12, pp.1909-1914, 2010.

[6] D. Wang, L. Zhang, and A. Vincent, "Motion Compensated Frame Rate Up-Conversion-Part I: Fast Multi-Frame Motion Estimation," IEEE Transactions on Broadcasting, vol.56, no.02, pp.133-141, 2010.

[7] B. Choi, J. Han, C. Kim, and S. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," *IEEE Transactions on Circuits Syst. Video Technology*, vol.17, no.04, pp.407–416, 2007.

[8] D. Wang, A. Vincent, P. Blanchfield, and R. Klepko, "Motion Compensated Frame Rate Up-Conversion-Part II: New Algorithms for Frame Interpolation," *IEEE Transactions on Broadcasting*, vol.56, no.02, pp.142-149. 2010.

[9] U. Kim, M. Sunwoo, "New Frame Rate Up-Conversion Algorithms with Low Computational Complexity," *IEEE Transactions on Circuits System Video Technology*, vol.24, no. 3, pp. 384-393, 2014.

[10] R. Salakhutdinov and G. E. Hinton. "Deep Boltzmann machines," *In Proc. of the International Conference on Artificial Intelligence and Statistics(AISTATS)*, vol.12, 2009.

[11] Y. LeCun L. Jackel, B Boser, J. Denker, "Handwritten Digit Recognition: Applications of Neural Net Chips and Automatic," *Neurocomputing: Algorithms, Architectures and Applications*, vol.68, pp.303-318, 2012.

[12] D. Eigen, D. Krishnan, R. Fergus, "Restoring An Image Through a Window Covered with Dirt or Rain," *in Proc. of International Converence on Computer Vision (ICCV)*, pp.633-640, 2013.

[13] C. Dong, C. C. Loy, K. He, X. Tang, "Learning a Deep Convolutional Network for Image Super-Resolution," *in Proc. of European Conference on Computer Vision (ECCV 14)*, pp.1-16. 2014.

[14] H. Burger, C. Schuler, S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D," *in Proc. of Computer Vision and Pattern Recognition (CVPR)*, pp. 2392–2399, *2012*.