SUPER-RESOLUTION OF COMPRESSED VIDEOS USING CONVOLUTIONAL NEURAL NETWORKS

Armin Kappeler *Seunghwan Yoo* *Qiqin Dai* *Aggelos K. Katsaggelos*

Northwestern University
Departement of EECS
Evanston, IL 60208, USA

ABSTRACT

Convolutional neural networks (CNN) have been successfully applied to image super-resolution (SR) as well as other image restoration tasks. In this paper, we consider the problem of compressed video super-resolution. Traditional SR algorithms for compressed videos rely on information from the encoder such as frame type or quantizer step, whereas our algorithm only requires the compressed low resolution frames to reconstruct the high resolution video. We propose a CNN that is trained on both the spatial and the temporal dimensions of compressed videos to enhance their spatial resolution. Consecutive frames are motion compensated and used as input to a CNN that provides super-resolved video frames as output. Our network is pretrained with images, which significantly improves the performance over random initialization. In extensive experimental evaluations, we trained the state-of-the-art image and video super-resolution algorithms on compressed videos and compared their performance to our proposed method.

Index Terms— Deep Learning, Convolutional Neural Networks, Super-Resolution, Video Compression

1. INTRODUCTION

Image and video or multiframe super-resolution is the process of estimating a high resolution version of a low resolution image or video sequence. It has been studied for a long time, but has become more prevalent with the new generation of Ultra High Definition (UHD) TVs. Most video content was not recorded in UHD resolution, therefore SR algorithms are needed to generate UHD content from Full HD (FHD) or lower resolutions.

SR algorithms can be divided into two categories, model-based and learning-based algorithms. Model-based approaches [1–4] model the Low Resolution (LR) image as a blurred, subsampled version of the High Resolution (HR) image with additive noise. The reconstruction of the HR image from the LR image is an ill-posed problem and therefore needs to be regularized. In a Bayesian framework, priors controlling the smoothness or the total variation of the image are introduced in order to obtain the reconstructed HR image. Learning-based algorithms learn representations (i.e. dictionaries) from large training databases of HR and LR image pairs [5, 6] or exploit self-similarities within an image [6, 7]. Dictionary based approaches utilize the assumption that natural image patches can be represented as a linear combination of learned dictionary patches or atoms. Yang *et al.* [5] were among the first to use two coupled dictionaries to learn a nonlinear mapping between the LR and the HR images.

In [8, 9], super-resolution has been applied to compressed video streams. Fully resolved keyframes are embedded into the video stream and utilized to enhance the remaining frames. The authors of [3, 10, 11] introduced methods that do not require keyframes. Their methods model the quantization error and use a Bayesian approach to reconstruct the high resolution frames. In the proposed method, the quantization error is incorporated in the training process, as we train the CNN on compressed frames.

Inspired by the recent successes achieved with CNNs [12, 13], a new generation of image SR algorithms based on deep neural nets emerged [14–19], with very promising performances. The training of CNNs can be done efficiently by parallelization using GPU-accelerated computing. Neural networks are capable of processing and learning from large training databases such as ImageNet [20], while training a dictionary on a dataset this size can be challenging. Moreover, once a CNN is trained, super-resolving an image is a purely feed-forward process, which makes CNN based algorithms much faster than traditional approaches. In this paper, we introduce a CNN framework for video SR.

In the classification and retrieval domains, CNNs have been successfully trained on video data [21, 22]. Training for recovery purposes remains a challenging problem because the video quality requirements for the training database are high since the output of the CNN is the actual video rather than just a label. Suitable videos for the SR task are uncompressed, feature-rich and should be separated by shots/scenes. We show that by pretraining the CNN with images we can bypass the creation of a large video database. Our proposed algorithm requires only a small video database for training to achieve very promising performance.

In this paper we introduce a compressed video SR framework based on CNNs. We propose a pretraining procedure whereby we train the pretraining SR architecture on images and utilize the resulting filter coefficients to initialize the training of the video SR architectures. This improves the performance of the video SR architecture significantly. In order to handle fast moving objects and motion blur in videos, we apply an adaptive motion compensation scheme.

The remainder of the paper is organised as follows. In Section 2 we explain our proposed framework. Sections 3 contains our results and experimental evaluation and Section 4 concludes the paper.

2. PROPOSED SUPER-RESOLUTION ALGORITHM

2.1. Image Pretraining

Before we start the training of the video SR model, we pre-train the model weights on images. For the image pretraining, we use a model for image SR, henceforth referred to as a pretraining model, with the network architecture parameters proposed in [15]. It has only convolutional layers which has the advantage that the input images

can be of any size and the algorithm is not patch-based. It consists of three convolutional layers, where the two hidden layers $H1$ and $H2$ are followed by a Rectified Linear Unit (ReLU) [23]. The first convolutional layer consists of $1 \times f_1 \times f_1 \times C_1$ filter coefficients, where $f_1 \times f_1$ is the kernel size and C_1 the number of kernels in the first layer. We use this notation to indicate that the first dimension is defined by the number of input images, which is 1 for the image SR case. The filter dimensions of the second and third layers are $C_1 \times f_2 \times f_2 \times C_2$ and $C_2 \times f_3 \times f_3 \times 1$, respectively. The last layer can only have one kernel in order to obtain an image as output. Otherwise an additional layer with one kernel otherwise a postprocessing or aggregation step is required. The input image is bicubically upsampled so that the input (LR) and output (HR) images have the same resolution. This is necessary because upsampling with standard convolutional layers is not possible. The model is trained on patches extracted from images from the ImageNet detection dataset [24], which consists of around 400,000 images.

2.2. Video Super-Resolution Architectures

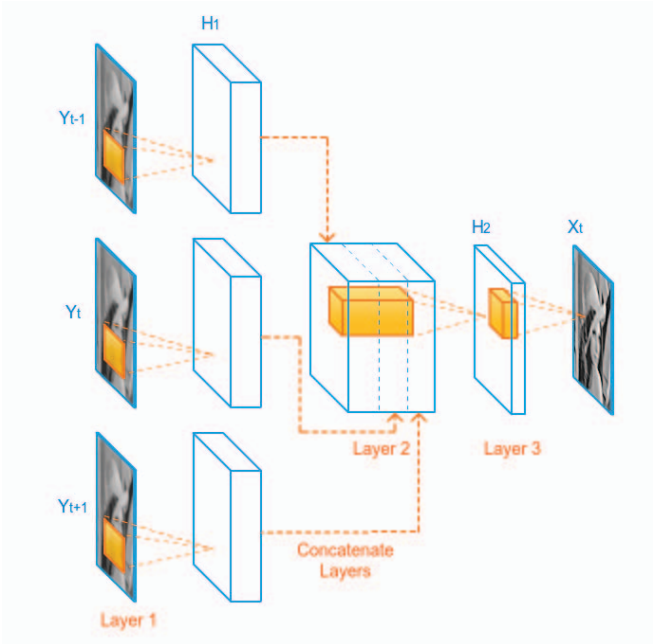

Fig. 1: Video SR architectures: The three input frames are concatenated (Concat Layer) before layer 2 is applied

It has been shown for model-based approaches that including neighboring frames into the recovery process is beneficial for video SR [2,3]. The motion between frames is modeled and estimated during the recovery process and additional information is gained due to the subpixel motions among frames. The additional information conveyed by these small frame differences can also be captured by a learning-based approach, if multiple frames are included in the training procedure. For the video SR architecture, we include the neighboring frames into the process. Figure 1 shows how we incorporate the previous and next frames into the process. For simplicity, we only show the architecture for three input frames, namely the previous (Y_{t-1}), current (Y_t), and next (Y_{t+1}) frames. Any number of past and future frames can be accomodated by adding more branches to the architecture (for example, we use five input frames in the experimental section). A single input frame has dimensions $1 \times M \times N$, where M and N are the width and height of the input image, respectively. The three input frames are first processed by layer 1 and then concatenated along the first dimension before the second and third convolutional layers are applied. The new input data for layer 2 is three times larger than in the pretraining SR architecture. Not only the data size but also the filter dimensions are larger for the video SR architectures. The filter coefficients of layer 2 increase to $3C_1 \times f_2 \times f_2 \times C_2$, whereas layers 1 and 3 remain the same. In order to transfer the filter values from the pretraining model to the video model, the kernel width, height and their number have to be equal in both models. The filters of layers 1 and 3 have the same dimensions and can be transferred directly from the pretraining model. The filter dimension of the second layer are different. The first dimension in the video model is three times larger than in the pretraining model, as the output data from the three filters from layer 1 are concatenated along the temporal dimension. Furthermore, the output data of layer 2 should be similar to the output data obtained by the single frame SR, as layers 1 and 3 remain the same as in the single frame SR. We transfer the video filter weights $w(.)$ and the biases $b(.)$ from the second layer of the pretraining model to the weights and biases from the video model $w_v(.)$ and $b_v(.)$ as

$$\begin{aligned} w_v(m,n,t-1,c) &= w_v(m,n,t,c) = w_v(m,n,t+1,c) \\ &= \frac{1}{3} w(m,n,t,c) \\ b_v(c) &= b(c), \qquad \forall m,n,c \end{aligned} \tag{1}$$

which is equivalent to averaging the input images before applying the second convolution layer.

2.3. Adaptive Motion Compensation

We tested a number of optical flow estimation algorithms [25]. Both, accuracy of the motion estimates and speed of implementation were taken into account. We chose Druleas algorithm [26] for our framework. The algorithm uses a Combined Local-Global approach with Total Variation (CLG-TV) and demonstrates good results even when large displacements are present.

Motion compensation can be difficult if large motion or motion blur occurs in the video. This can lead to undesired boundary effects and artifacts in the HR reconstruction and will therefore reduce performance. We propose the use of an adaptive motion compensation (AMC) scheme that reduces the influence of neighboring frames for the reconstruction in case of misregistration. Motion-compensation is applied according to the following equation

$$y^{amc}_{t-T}(i,j) = (1 - r(i,j)) y_t(i,j) + r(i,j) y^{mc}_{t-T}(i,j), \tag{2}$$

where $r(i,j)$ controls the convex combination between the reference and the neighboring frame at each pixel location (i,j), y_t is the center frame, y^{mc}_{t-T} is the motion compensated neighboring frame and y^{amc}_{t-T} is the neighboring frame after applying adaptive motion compensation. Similarly to [27], $r(i,j)$ is defined as

$$r(i,j) = exp(-ke(i,j)), \tag{3}$$

where k is a constant parameter and $e(i,j)$ is the motion compensation or misregistration error. Large errors can occur due to large motion, occlusion, blurring of the object, or due to the fact that (i,j) is close to a motion boundary. Using the adaptive motion compensation helped improve the performance for challenging videos, as will be shown in the experimental section.

Table 1: Average PSNR values for the *Myanmar* test sequences. The result of the proposed method (CVSRnet) is shown in the last column

		Single Frame SR Algorithms				Video SR Algorithms				Own
Scale	CRF	Bicubic	SrSC [5]	A+ [6]	SRCNN [15]	ANN [18]	Bayesian [2]	Bayesian-MB [4]	Enhancer [28]	CVSRnet
2	0	34.59	36.36	37.19	37.79	35.18	35.56	36.41	35.94	**38.48**
2	20	33.55	34.82	35.14	35.07	34.06	34.38	34.73	34.25	**36.06**
2	30	31.64	32.04	32.32	32.42	31.82	31.96	32.07	31.84	**32.81**
2	40	28.39	28.38	28.49	28.80	28.38	28.35	28.38	28.40	**28.95**
3	0	31.59	32.71	33.48	33.88	32.55	32.20	32.74	32.50	**34.42**
3	20	31.02	32.07	32.46	32.52	31.72	31.07	31.77	31.59	**33.14**
3	30	29.64	30.08	30.35	30.55	29.90	29.47	29.82	29.89	**30.82**
3	40	26.90	26.92	27.07	27.33	26.75	26.58	26.73	26.95	**27.39**

3. EXPERIMENTAL SECTION

3.1. Datasets

We used a publicly available video database [29] which provides high quality movies. The *Myanmar* video sequence was used for the training and testing of our algorithm. The video contains 59 scenes from which we use 53 for training and 6 for testing. We use 4 frames from each test sequence and calculate the average PSNR values from the 24 (6×4) resulting test frames as a performance measure. The *Myanmar* video is uncompressed and has 4K resolution (3840×2160 pixels). We downsampled the video to a 960×540 pixel resolution (HR), in order to better compare to the state-of-the-art SR algorithms. We used ffmpeg [30] to create the LR videos. We first downsampled and then compressed the video with the H.264 codec. For our experiments, we choose four different compression levels using the constant rate factor (CRF) values 0, 20, 30, and 40, where 0 is lossless and 40 is the worst quality.

3.2. Implementation

We implemented our proposed algorithm with the Caffe framework [31]. The pretraining network has 3 convolutional layers, where layers 1 and 2 are each followed by a ReLU. Layer 1 has 64 kernels with a kernel size of 9x9, Layer 2 has 32 kernels with size 5x5 and the third layer has one 5x5 kernel [15]. The filters of the video SR architectures have the same configurations as the pretraining SR architecture. Following the literature ([5, 15]), we converted the images and videos into the YCbCr colorspace and only used the luminance channel (Y) for training, testing, and PSNR calculation. For the color images shown in this paper, we bicubically upsampled the chrominance channels, Cb and Cr. In order to create the video training set, we extracted sets of 5 consecutive frames from the HR and the compressed LR training video scenes. Then, we upsampled the LR video frames by the desired factors with bicubic interpolation to their original resolution using the Matlab implementation of *imresize*. Afterwards, we calculated the optical flow from the first and the last two frames towards the center frame and computed the motion compensated frames. From the resulting 5 frames (4 motion compensated and one center frame) we extracted $36 \times 36 \times 5$ data cubes, that is, 36×36 pixel patches from 5 consecutive frames. We dismissed patches/data cubes if they did not contain sufficient structures. The created training database consists of about 900,000 data cubes. We used the Euclidean distance between the output image and the ground truth image as loss function. In order to avoid border effects during the training, we only use the 20×20 center pixels of the original patch to calculate the Euclidean loss. The model weights provided by SRCNN [15] were used for the image SR pretraining. All the results shown in the experimental Section are evaluated after 200,000 iterations if pretraining was used, and 400,000 iterations otherwise.

3.3. Comparison to the State-of-the-Art

We compare our algorithm, henceforth referred to as Compressed Video SR network (CVSRnet), to the state-of-the-art single frame and video SR algorithms. The implementations and parameters of dictionary-based Sparse Coding SR (ScSR) [5], adjusted anchored neighbor regression (A+) [6] Enhancer [28] and Bayesian-MB [4] were used as provided by the authors. For the Bayesian adaptive video SR method from [2] (Bayesian) we used the reimplementation by [4]. We reimplemented SRCNN [15] and ANN[1] [18] using the Caffe framework [31]. We trained new dictionaries for ScSR and A+ and new models for SRCNN and ANN on the same compressed video database that we used for the training of our own method (CVSRnet). All video SR methods were tested using $\pm$ 2 neighboring frames.

Table 1 shows the average PSNR values for the different algorithms tested on the *Myanmar* sequence. The proposed CVSRnet provides the highest average PSNR for both upscale factors and all four compression rates. Figure 2 shows an example frame from the *Myanmar* test set using upscale factor 3. We show the original frame, the result obtained by bicubic interpolation, the second best result by SRCNN [15] and our proposed algorithm (CVSRnet). In addition for the second row, we show the results that we obtained when we trained a model on uncompressed videos (CVSRnet-CRF0) and used it for the super-resolution of the with CRF 30 compressed test frame. We can see that CVSRnet-CRF0 created artifacts around the edges for the CRF 30 compressed frame because the training and test data are not consistent. We can also see an artificial vertical transition line on the tower created by SRCNN whereas the tower in the CVSRnet image was reconstructed smoothly.

3.4. Execution Time

We trained and tested the proposed architecture with 3 and 5 input frames for the Myanmar sequence. For 3 frames we need 14 hours for the training and achieve a PSNR of 34.33 dB. 5 input frames require 22 hours for a PSNR of 34.42 dB. Spending 8 hours more training time for 0.09 dB seems a rather large effort, however the training happens in advance and is not time critical. The runtime of the super-resolution process is mainly determined by the motion compensation, which takes about 55 second per frame. The runtime of the CNN is 0.17 seconds for 3 input frames and 0.24 seconds for 5 input frames, and is therefore neglectable.

3.5. Pretraining

In this section we trained two models for upscale factor 3 with 3 input frames and no compression to demonstrate the importance of

[1] we used ReLU instead of Sigmoid as activation function, and bicubically upsampled input frames instead of the original LR frames. Both changes led to a better PSNR.

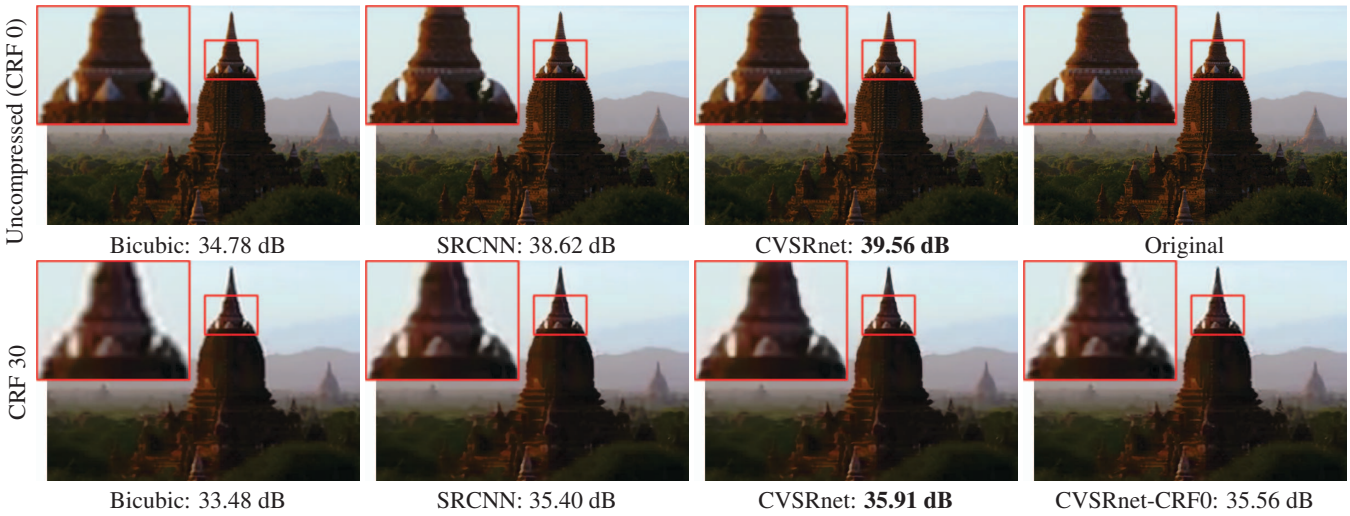

Fig. 2: Comparison to the State-of-the-Art: SR frames from the Myanmar video compared to our method (CVSRnet) for upscale factor 3

Fig. 3: Frame with motion blurred pigeon from the *walk* sequence, reconstructed with MC and AMC for upscale factor 3

pretraining. The filters of the model without pretraining was initialized with random Gaussian distributed values with standard deviation of 0.001 and was trained for 400,000 iterations. The pretrained model was initialized with the filter weights from the pretraining SR model. The performance of this model did not improve any further after 200,000 iterations due to the pretrained values. Even without pretraining we achieve a PSNR of 33.71 dB which is comparable to the state-of-the-art. However using pretraining leads to a PSNR of 34.33 dBs which is an improvement of 0.62 dBs.

3.6. Adaptive Motion Compensation

We compared the CVSRnet models with and without motion compensated frames to show the effect of motion compensation (MC) and adaptive motion compensation (AMC). We can see in Table 2 that MC performs best and improves the average PSNR by 0.16 dBs over no MC and 0.17 dBs over AMC. By applying motion compensation, the neighboring frames become very similar to the center frame. The remaining differences between the reference and the motion compensated frames contain information that is beneficial for the video SR learning process. In order to verify this, we trained a model where we replaced the 5 consecutive input frames with the center frame. Therefore for this training there is no temporal information available (No Temp.). This model is comparable to the pretraining architecture but with 5 times more filter coefficients. The PSNR of this model is 0.43 dB below the MC result. Although the average PSNR of AMC on our test videos is lower than with normal MC, the use of AMC led to significant improvements on frames with strong motion blur and fast moving objects. Figure 3 shows a frame from the *walk* sequence with a flying pigeon. The wings experience strong motion blur due to their fast motion. We tested the frames using the proposed CVSRnet algorithm with AMC and standard MC, where we set the constant k from Equation 3 to $1/8$. In addition, we tested the Bayesian-MB method [4], which is designed to handle motion blurred frames. The standard MC approach fails to estimate the motion of these objects which leads to a poor SR reconstruction. Even the Bayesian-MB method produces artifacts in the shape of small dots in both examples. The AMC on the other hand successfully reconstructs the pigeon and improves the PSNR by 0.27 dBs.

Table 2: PSNR for no motion compensation (MC), no temporal information (No Temp), normal MC and adaptive MC (AMC) for upscale factor of 3 for the Myanmar video.

No MC	No Temp.	MC	AMC
34.26	33.99	**34.42**	34.25

4. CONCLUSION

In this paper we have introduced a compressed video SR algorithm using convolutional neural networks. Our proposed CNN exploits spatial as well as temporal information. To the best of our knowledge, there is no existing work on CNN based compressed video SR. Using motion compensated input frames and a pretraining method we were able to improve the reconstruction quality and reduce the training time. Finally, we introduced an adaptive motion compensation scheme to deal with motion blur and fast moving objects. We presented an algorithm that outperforms the current state-of-the art algorithms.

5. REFERENCES

[1] S D Babacan, R Molina, and A K. Katsaggelos, "Variational Bayesian Super Resolution.," *IEEE Transactions on Image Processing*, vol. 20, no. 4, pp. 984–999, 2011.

[2] Ce Liu and Deqing Sun, "On Bayesian Adaptive Video Super Resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 346–360, 2014.

[3] A. K. Katsaggelos, R. Molina, and J. Mateos, *Super Resolution of Images and Video*, San rafael edition, 2007.

[4] Z. Ma, J. Jia, and E. Wu, "Handling Motion Blur in Multi-Frame Super-Resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, vol. 1.

[5] J Yang, J. Wright, T.S. Huang, and Y Ma, "Image Super-Resolution Via Sparse Representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[6] R Timofte, V De Smet, and L Van Gool, "A+: Adjusted Anchored Neighborhood Regression for Fast Super-Resolution," *IEEE Asian Conference on Computer Vision*, pp. 1920–1927, 2014.

[7] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja, "Single Image Super-Resolution From Transformed Self-Exemplars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206.

[8] F Brandi, R de Queiroz, and D Mukherjee, "Super-resolution of video using key frames and motion estimation," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 321–324.

[9] A Mahfoodh, D Mukherjee, and H Radha, "Super-resolution for inconsistent scalable video streaming," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3019–3023.

[10] X Zhang, M Tang, and R Tong, "Robust super resolution of compressed video," *The Visual Computer*, vol. 28, no. 12, pp. 1167–1180, 2012.

[11] C A Segall, A K Katsaggelos, R Molina, and J Mateos, "Bayesian resolution enhancement of compressed video," *Image Processing, IEEE Transactions on*, vol. 13, no. 7, pp. 898–911, 2004.

[12] A Krizhevsky, I Sutskever, and G E Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[13] C Szegedy, W Liu, Y Jia, P Sermanet, S Reed, D Anguelov, D Erhan, V Vanhoucke, and A Rabinovich, "Going deeper with convolutions," *arXiv preprint:1409.4842*, 2014.

[14] A Kappeler, S Yoo, Q Dai, and A K Katsaggelos, "Video super-resolution with convolutional neural nets," *to appear in Computational Imaging, IEEE Transactions on*, 2016.

[15] C Dong, C C Loy, K He, and X Tang, "Learning a Deep Convolutional Network for Image Super-Resolution," in *Proceedings of the IEEE European Conference on Computer Vision*, 2014.

[16] Z Cui, H Chang, S Shan, B Zhong, and X Chen, "Deep Network Cascade for Image Super-Resolution," *Proceedings of the IEEE European Conference on Computer Vision*, pp. 1–16, 2014.

[17] Z Wang, Y Yang, Z Wang, S Chang, W Han, J Yang, and T S Huang, "Self-Tuned Deep Super Resolution," *arXiv preprint arXiv:1504.05632*, 2015.

[18] M. Cheng, N. Lin, K. Hwang, and J. Jeng, "Fast video super-resolution using artificial neural networks," *2012 8th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, pp. 1–4, 2012.

[19] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, "Video Super-Resolution via Deep Draft-Ensemble Learning," *ICCV*, 2015.

[20] J Deng, W Dong, R Socher, and Lj Li, "A large-scale hierarchical image database," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[21] A Karpathy and G Toderici, "Large-scale video classification with convolutional neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.

[22] S Ji, W Xu, M Yang, and K Yu, "3D Convolutional Neural Networks for Human Action Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[23] Al Maas, Ay Hannun, and Ay Ng, "Rectifier nonlinearities improve neural network acoustic models," *ICML*, vol. 28, 2013.

[24] O Russakovsky, J Deng, H Su, J Krause, S Satheesh, S Ma, Z Huang, A Karpathy, C. V. Jan, J. Krause, and S. Ma, "ImageNet Large Scale Visual Recognition Challenge," *arXiv preprint arXiv:1409.0575*.

[25] S Baker, D Scharstein, J. P. Lewis, S Roth, M J. Black, and R Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[26] M Drulea and S Nedevschi, "Total variation regularization of local-global optical flow," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 318–323, 2011.

[27] M. K. Park, M. G. Kang, and A. K Katsaggelos, "Regularized high-resolution image reconstruction considering inaccurate motion information," *Optical Engineering*, vol. 46, no. 11, pp. 117004, 2007.

[28] Infognition, "Video Enhancer, "http://www.infognition.com/videoenhancer"," 2010.

[29] "Harmonic Inc, "http://www.harmonicinc.com/resources/videos/4k-video-clip-center"," 2014.

[30] ffmpeg, "ffmpeg, "http://www.ffmpeg.org"," 2010.

[31] Y Jia, E Shelhamer, J Donahue, S Karayev, J Long, R Girshick, S Guadarrama, and T Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," *arXiv preprint arXiv:1408.5093*, 2014.