CS230

Up-sampling low resolution surveillance feeds using GANs

Abdalla Al-Ayrot
Stanford University
abdalla@stanford.edu

Jake Sparkman
Stanford University
jdspark@stanford.edu

Utkarsh Contractor
Stanford University
utkarshc@stanford.edu

1 Introduction and Motivation

There has been much work done on video and image super resolution, where previous approaches have handled up-sampling of both images and videos. However, there is less literature on up-sampling images and videos in the surveillance domains, especially up-sampling low resolution security footage. We propose using Generative Adversarial Networks (GANs) to upscale low resolution surveillance videos. The outcome will be a model that will take as input a low resolution video (surveillance) and output a video at a higher resolution. This video super resolution output can be used by both manual surveillance operators as well as machine learning algorithms both in batch and real time scenarios to identify and analyze anomalous events in security feeds. A further extension of this work could be used for both scene and subject restoration in the surveillance videos.

2 Dataset

There exist a number of low resolution video anomaly datasets used by the deep learning community for research. However few surveillance video datasets exist with HR videos. Our focus in this study will be on up-scaling surveillance videos from the UCF Video Anomaly Detection Dataset[19] so we can provide the research community with a high quality HR video anomaly dataset. We use the Video Anomaly Detection Dataset [19] for training, validation and testing. The original dataset consists of 128 hours of videos consisting of 1900 real-world surveillance videos, with 13 realistic anomalies such as fighting, road accident, burglary, robbery, etc. as well as normal activities captured by surveillance cameras. The videos are long untrimmed surveillance videos with low resolution with large intra-class variations due to changes in camera viewpoint and illumination, and background noise, thus making it a great fit for our use case. For our experiments we use a miniature version of this dataset with 2207 images, split into 1200 training, 507 validation images and 500 images in the test set. Few samples from the dataset can be seen in Figure 1.

3 Approach and Methodology

3.1 Choice of Neural Architecture

Super-resolution (SR) is used to upscale an image or video from a low-resolution (LR) image or video to a high-resolution (HR) image or video. The method is essentially estimating the HR image or video from its LR equivalent.

CS230: Deep Learning, Winter 2021, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

Figure 1: Sample frames from Video Anomaly Detection Dataset [19]

Advances in the research of learning-based methods, have increased the performance of single-image SR (SISR) significantly. Applying the advances in SISR to video SR (VSR) however is more challenging, as simply taking SISR over each video frame, can lead to inferior results due to temporal coherency issues. Recent research has shown that combining state-of-the-art SISR and (multi image SR) MISR methods with a spatio-temporal approach surpasses state-of-the-art VSR results with the introduction of iSeeBetter.[2] iSeeBetter essentially uses recurrent-back-projection networks as its generator to extracts spatial and temporal information from prior, current and next frames. For it to improve the "naturally" of the super-resolved images the generator from super-resolution generative adversarial network (SRGAN) is used.

To this effect, we have chosen to experiment with the following approaches:

- Baseline Approach: Sub-pixel CNN Based Video Up-sampling
- Proposed Approach 1: Enhanced SRGAN Based Video Up-sampling
- Proposed Approach 2: Recurrent Back-Projection Networks (RBPN) + SRGAN Based Video Up-sampling

3.2 Baseline Approach - Sub-pixel CNN Based Architecture

3.2.1 Model Details and Architecture

Our baseline approach follows work done by Shi et al. on Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network [17] which is the first convolutional neural network (CNN) capable of real-time SR of 1080p videos on a single K2 GPU. In this approach, contrary to previous works, the authors propose to increase the resolution from LR to HR only at the very end of the network and super-resolve HR data from LR feature maps. This eliminates the need to perform most of the SR operation in the far larger HR resolution. For this purpose, this approach proposes an efficient sub-pixel convolution layer to learn the upscaling operation for image and video super-resolution. For the baseline version architecture we will be using l = 3, (f1, n1) = (5, 64), (f2, n2) = (3, 32) and f3 = 3 in our evaluations. The choice of the parameter is inspired by SRCNN's [5] 3 layer 9-5-5 model.

3.2.2 Training and Scoring Details

In the training phase, 17r × 17r pixel sub-images are extracted from the training ground truth images IHR, where r is the upscaling factor. To synthesize the low-resolution samples ILR, we blur IHR using a Gaussian filter and sub-sample it by the upscaling factor. The sub-images are extracted from

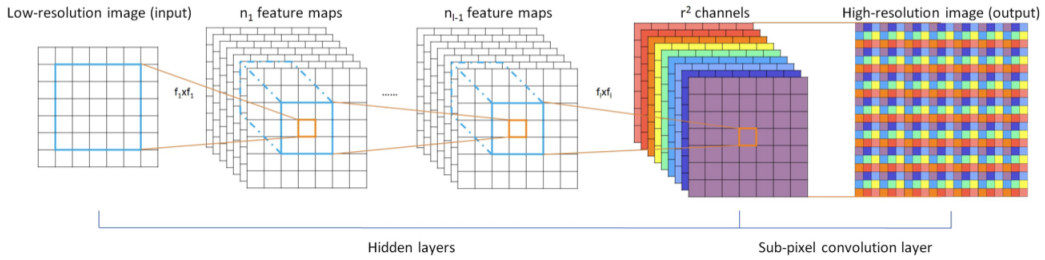


Figure 2: The proposed efficient sub-pixel convolutional neural network (ESPCN), with two convolution layers for feature maps extraction, and a sub-pixel convolution layer that aggregates the feature maps from LR space and builds the SR image in a single step.

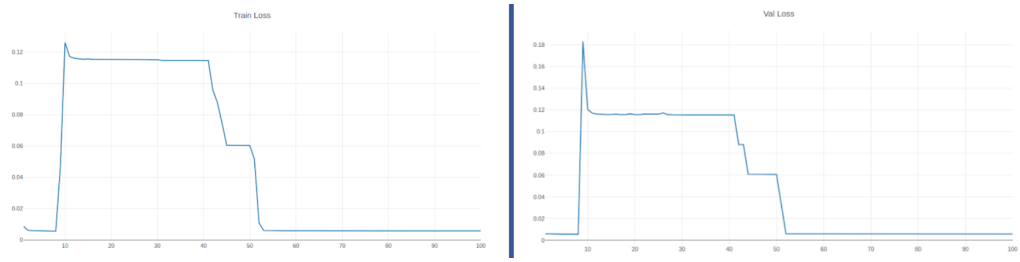


Figure 3: Train and Validation loss for sub-pixel convolutional neural network (ESPCN).

original images with a stride of

$$(17 \sum mod(f, 2)) * r$$

from IHR and a stride of

$$17 \sum mod(f, 2)$$

from ILR. This ensures that all pixels in the original image appear once and only once as the ground truth of the training data. The training stops after no improvement of the cost function is observed after 100 epochs. Initial learning rate is set to 0.01 and final learning rate is set to 0.0001 and updated gradually when the improvement of the cost function is smaller than a threshold μ. The training takes roughly about 12 hours on a 1080 Titan X Pascal GPU on images from UCF Mini Image Dataset [20] for upscaling factor of 4 and 100 epochs. The train and validation charts can been seen in Figure 3.

3.3 Proposed Approaches: GAN Based Up-Sampling

3.3.1 Enhanced SRGAN - Model Details and Architecture

Our first proposed approach is inspired by Ledig et al. on Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network [14]. This was first framework capable of inferring photo-realistic natural images for 4x upscaling factors using a deep residual network to recover photo-realistic textures from heavily down-sampled images. In this approach a super-resolution generative adversarial network (SRGAN) for which is applied a deep residual network (ResNet) with skip-connection and diverge from MSE as the sole optimization target. Different from previous works, this approach defines a novel perceptual loss using high-level feature maps of the VGG network[18] combined with a discriminator that encourages solutions perceptually hard to distinguish from the HR reference images.

At the core of this very deep generator network G, which is illustrated in Figure 3 are B residual blocks with identical layout. Inspired by Johnson et al. [12] employed are the block layout. Specifically, two convolutional layers with small 3×3 kernels and 64 feature maps followed by batch-normalization layers and ParametricReLU [10] as the activation function. The resolution of the input image is increased with two trained sub-pixel convolution layers as proposed by Shi et al. [17]. To discriminate real HR images from generated SR samples a discriminator network is trained. The architecture is shown in Figure 4. The architectural guidelines followed are summarized by Radford et al.

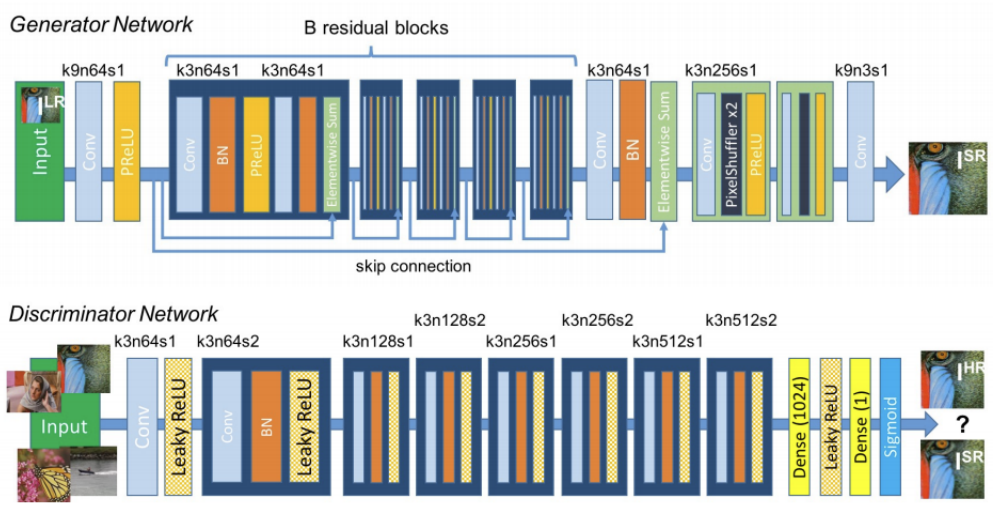


Figure 4: Architecture of Generator and Discriminator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

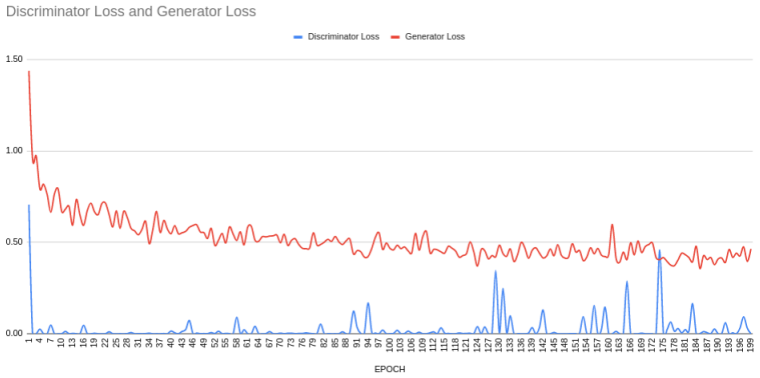


Figure 5: GAN losses for Enhanced SRGAN training

[16] and use LeakyReLU activation (= 0.2) and avoid max-pooling throughout the network. The discriminator network contains eight convolutional layers with an increasing number of 3 × 3 filter kernels, increasing by a factor of 2 from 64 to 512 kernels as in the VGG network [18]. Strided convolutions are used to reduce the image resolution each time the number of features is doubled. The resulting 512 feature maps are followed by two dense layers and a final sigmoid activation function to obtain a probability for sample classification.

Training and Scoring Details For our v1 SRGAN version we have run the model on our UCF[20] mini dataset. All experiments are performed with a scaling factor of ×4 between LR and HR images. We obtain LR images by down-sampling HR images using the MATLAB bicubic kernel function. The mini-batch size is set to 16. The spatial size of cropped HR patch is 128 × 128. Training a deeper network benefits from a larger patch size, since an enlarged receptive field helps to capture more semantic information. However, it costs more training time and consumes more computing resources. The training process is divided into two stages. First, train a PSNR-oriented model with the L1 loss. , then employ the trained PSNR-oriented model as an initialization for the generator. The learning rate is set to 0.0001 and halved at [10, 30 , 50, 75] iterations.

Pre-training with pixel-wise loss helps GAN-based methods to obtain more visually pleasing results. The reasons are that 1) it can avoid undesired local optima for the generator; 2) after pre-training, the discriminator receives relatively good super-resolved images instead of extreme fake ones (black

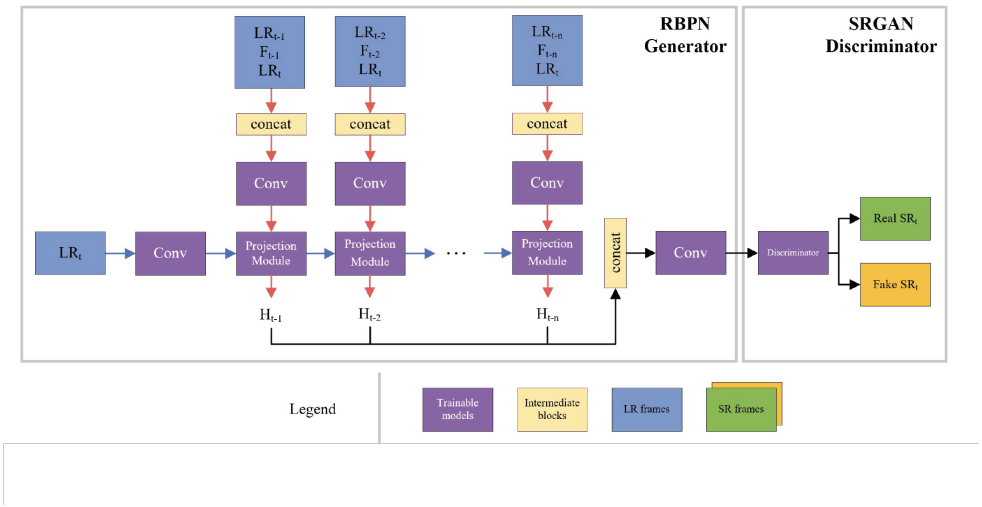


Figure 6: Overview of iSeeBetter [2]

Table 1: Adopted notation for iSeeBetter [2]

HR_t	input high resolution image
LR_t	low resolution image (derived from HR_t)
F_t	optical flow output
H_{t-1}	residual features extracted from (LR_{t-1}, F_{t-1}, LR_t)
SR_t	estimated HR output

or noisy images) at the very beginning, which helps it to focus more on texture discrimination. For optimization, we use Adam with 1 = 0.9, 2 = 0.999. We implement our models with the PyTorch framework and train them using NVIDIA Titan X Pascal GPUs. The SRGAN training is very computational expensive and takes about 36 hours on a Titan X Pascal 1080 GPU for 200 EPOCHS for 1707 images from UCF Mini dataset [20]. The Generator and Discriminator (GAN) losses are seen in Figure 5.

3.3.2 iSeeBetter - Model Details and Architecture

This approach is inspired by iSeeBetter [2]. The framework achieved state-of-the-art results by combining Recurrent Back-Projection Networks (RBPN) as its generator and the discriminator from SRGAN. Figure 7 shows the original architecture of the iSeeBetter-network (see Table 1 for adopted notation). The RBPN generator preserves spatio-temporal information by combining SISR and MISR. The horizontal flow of the network (illustrated by the blue lines in Figure 7) upsamples LR_t using SISR, with a DBPN architecture [9]. Up-down-up sampling is performed using 8 x 8 kernels with a stride of 4 and a padding of 2. As with enhanced SRGAN we use ParametricReLU [10] as the activation function. The vertical flow of the network (illustrated by the red arrows in Figure 7) performs MISR by utilizing a ResNet Architecture. We use three tiles of five blocks each consisting of two convolutional layers with 3 × 3 kernels, padding of 1 and stride of 1. As with enhanced SRGAN and the DBPN architecture we use ParametricReLU [10] as the activation function. The MISR computes the residual features from LR_t, its neighboring frames (LR_{t-1}, ..., LR_{t-n}) and the precomputed dense motion flow maps (F_{t-1}, ..., F_{t-n}).

RBPN detects missing information from LR_t at each projection stage and recovers details by extracting residual features from neighboring frames. As a result, the convolutional layers that feed the projection modules in Figure 7 act as feature extractors.

Training and Scoring Details We ran the model on our UCF [20] mini dataset for our v1 iSeeBetter edition. The LR and HR images are scaled by a factor of four in all experiments. We get LR images by using the MATLAB bicubic kernel function to downsample HR images. The size of the mini-batch

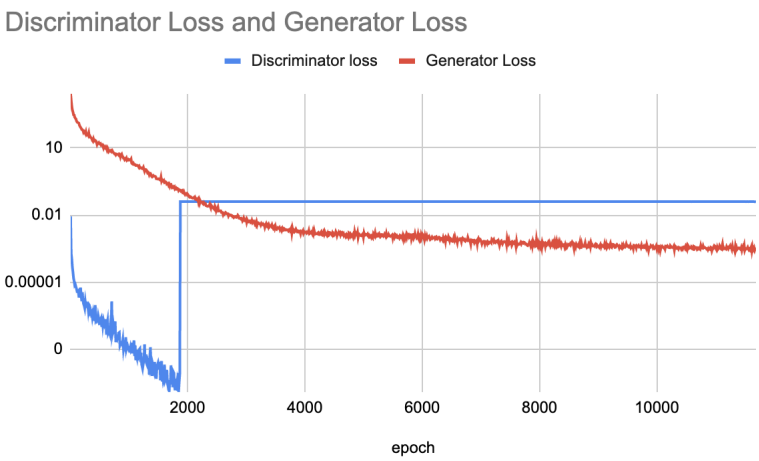


Figure 7: GAN losses for iSeeBetter training

Figure 8: Comparison of our model performance across various neural architectures.

is set to one. The cropped HR patch has a spatial size of 32x32. A larger patch size helps to train a deeper network since a larger receptive area helps to collect more semantic knowledge. However, it takes longer to practice and uses more computational power.

MSE is the most widely used loss function in a wide range of state-of-the-art SR methods that seek to increase an image's PSNR to determine image quality. [22] Optimizing for MSE during training is widely known to increase PSNR and SSIM. These metrics, however, may fail to capture fine details in the image, ensuing in a misrepresentation of perceptual quality. [14] The reason for this it was found in some experiments that some manually distorted images had an MSE score comparable to the original image.[2] We train the model with the four loss functions originally proposed in iSeeBetter (MSE, perceptual, adversarial, and TV) and weight the results for each frame. We use the PyTorch framework to build our models, and we train them on NVIDIA Tesla V100 GPUs. The UCF Mini dataset [20] was used to train the model. The Generator and Discriminator (GAN) losses are seen in Figure 6.

4 Evaluation Methodology/Results

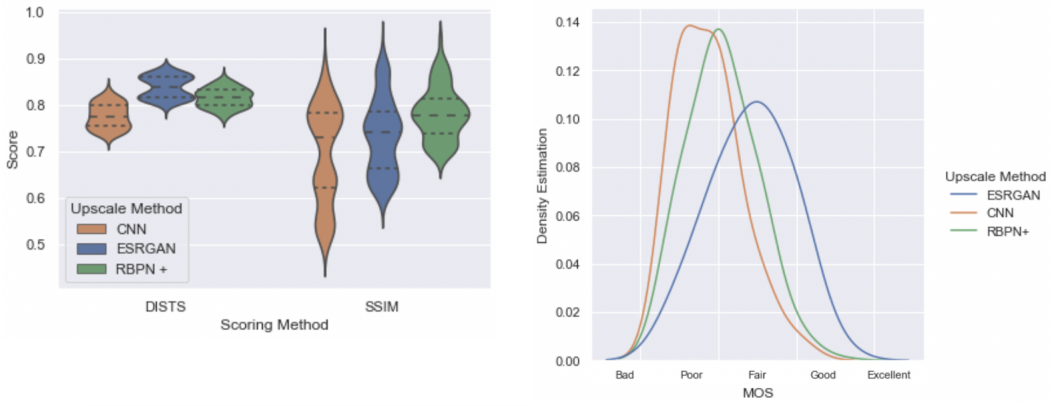
Our models performance has been quantified using several image quality assessment models. We built on the work of Ding et al. on the Comparison of Full-Reference Image Quality Models for Optimization of Image Processing Systems[13]. They found that the Deep Image Structure and Texture Similarity (DISTS) model was the most robust model evaluated, and was superior at evaluating super resolution tasks.

1. DISTS, or Deep Image Structure and Texture Similarity metric [4], is explicitly designed to tolerate texture resampling (e.g., replacing one patch of grass with another). DISTS is based on an injective mapping function built from a variant of the VGG network, and

combines SSIM-like structure and texture similarity measurements between corresponding feature maps of the two images. It is sensitive to structural distortions but at the same time robust to texture resampling and modest geometric transformations. The fact that it is robust to texture variance is also helpful when evaluating images generated by GANs. We used an open source implementation provided by Ding et al[13]. As DISTS maps images to a perceptual distance space, a score closer to 0 corresponds to more similar images. We have chosen to report scores as $1 - DISTS_{dist}$ to allow for an intuitive comparison. We used it as our objective measure of performance.

2. SSIM, or Structural Similarity (SSIM) index [24], has become a standard way of measuring IQA for a variety of applications. We included it for reference as many people are familiar with this method and it is used quite frequently. A score closer to one indicates a higher quality image.

3. MOS, or Mean Opinion Score, is the average of human judgements on the quality of the image. This metric is based entirely on human perception, and as such it is dependent on a variety of environmental factors that affect the display of the image. To compensate for human biases and differences in environment, we randomly selected 15 frames which we held constant across the upsampling methods. For each frame we sourced 15 separate individuals via Mechanical Turk to provide a MOS score for that frame, in isolation of any other images.



	ESRGAN		CNN		RBPN + SRGAN	
	DISTS	SSIM	DISTS	SSIM	DISTS	SSIM
Count	506	506	506	506	506	506
Mean	0.839	0.734	0.777	0.695	0.816	0.784
Std	0.976	0.081	0.973	0.105	0.979	0.057
Min	0.886	0.583	0.842	0.493	0.870	0.677
25%	0.861	0.664	0.800	0.623	0.832	0.739
50%	0.840	0.742	0.776	0.731	0.816	0.779
75%	0.818	0.787	0.755	0.784	0.799	0.814
Max	0.786	0.934	0.720	0.906	0.766	0.948

	ESRGAN MOS	CNN MOS	RBPN+ MOS
Count	225	225	225
Mean	2.83	1.75	2.09
Excellent	5	0	1
Good	57	7	9
Fair	81	27	56
Poor	58	93	103
Bad	24	98	56

Figure 9: DISTS, SSIM, and MOS Distributions

Using 506 images on the UCF dataset, we evaluated both our ESRGAN model and the RBPN+ models against the baseline. The results shown in Figure 9.

Our ESRGAN model was found to have both a higher DISTS and MOS score distribution. ESRGAN yielded over a full point increase in MOS mean compared to the baseline versus a third of a point increase with RBPN+. However, RBPN+ was found to have the highest SSIM score which again suggests that SSIM scores are not predictive of higher MOS scores, echoing the finding by Ding et al.[13]. We have also listed all details on our choice of hyper parameters, train/val losses and evaluation results in Figure 10.

	TRAINING DETAILS						EVALUATION / RESULTS		
Neural Architecture	Hyperparameters	# of epochs	Train Loss	Validation Loss	Train PSNR	Validation PSNR	MOS Mean (15 Samples, 15 Graders)	DSITS-Median	SSIM - Median
BASELINE - CNN Based	"Upscale factor = 4 CNN Architecture - 9-5-5 lr = 1e-2 gamma = 0.1 optimizer = adam batch size = 64"	100	0.0056	0.0057	22.46	22.49	1.75	0.78	0.73
			D-Loss	G-Loss	Content	ADV			
Enhanced SRGAN	"Upscale factor = 4 Neural Architecture - SRGAN lr = 1e-3 optimizer = adam batch size = 4 number of residual blocks = 30"	100	0.0042	0.0044	20.32	19.38	**2.83**	**0.84**	0.74
Recurrent Back-Projection Networks (RBPN) + SRGAN	"Upscale factor = 4 Neural Architecture - RBPN + SRGAN lr = 1e-4 batch size = 1 patch size = 32 number of frames = 9"	11359	0.0399	0.0003			2.09	0.82	**0.78**

Figure 10: Training and Scoring Hyper Parameters and Evaluation Metrics

5 Conclusion

We were able to experiment and evaluate three approaches for video up-sampling of surveillance videos on the UCF Mini [20] dataset. In conclusion we find that the Enhanced SRGAN based approach produce the highest quality and resolution for the 4x up-sampled images and videos across the three approaches we evaluated as listed in Section 3 (Approach and Methodology) as demonstrated in the Evaluation Methodology/Results section and as shown quantitatively by the SSIM and DISTS scores in Figure 8. Also in Figure 8 we see that the Mean Opinion Score (MOS) show consistency with our quantitative evaluation. Another important observation is the scoring latency of these approaches; The Sub-pixel CNN Based Video Up-sampling performs 4x up-scaling at the lower latency, almost 10x faster than the GAN based approaches that we have implemented. This makes the Sub-pixel CNN Based Video Up-sampling a good fit for more real time surveillance scenarios with a trade-off of up-scaling quality, whereas the GAN based approach produce higher quality up-scaling with high latency, making them a good fit for offline up-sampling scenarios.

6 Future Work

Through this work, we strive to provide a novel evaluation and adaptation of state of the art neural architectures to up-sample low resolution surveillance images and videos. Building on this work we plan to use our best performing model to up-sample the UCF Anomaly Videos [20] dataset to provide the research community a higher quality surveillance dataset to experiment on. We also plan to build upon our experiments to apply them in the domain of surveillance scene and subject restoration using object detection and up-sampling using our models.

References

[1] Marco Bevilacqua, A. Roumy, C. Guillemot, and M. Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *BMVC*, 2012.

[2] Aman Chadha, John Britto, and M. Mani Roja. iseebetter: Spatio-temporal video super-resolution using recurrent generative back-projection networks. *CoRR*, abs/2006.11161, 2020.

[3] Utkarsh Contractor, Chinmayi Dixit, and Deepti Mahajan. Cnns for surveillance footage scene classification, 2018.

[4] Wang S Simoncelli EP Ding K, Ma K. Image quality assessment: Unifying structure and texture similarity. https://arxiv.org/abs/2004.07728.

[5] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *CoRR*, abs/1501.00092, 2015.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[7] Fernando A. Fardo, Victor H. Conforto, Francisco C. de Oliveira, and Paulo S. Rodrigues. A formal evaluation of psnr as quality measurement parameter for image segmentation algorithms, 2016.

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[9] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution, 2018.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015.

[11] Tobias Hoßfeld, Poul E. Heegaard, Martín Varela, and Sebastian Möller. Qoe beyond the mos: an in-depth look at qoe via better metrics and their relation to mos. *Quality and User Experience*, 1(1), Sep 2016.

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution, 2016.

[13] Shiqi Wang Eero P. Simoncelli Keyan Ding1, Kede Ma1. Comparison of full-reference image quality models for optimization of image processing systems, 2020.

[14] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016.

[15] Jim Nilsson and Tomas Akenine-Möller. Understanding ssim, 2020.

[16] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[17] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *CoRR*, abs/1609.05158, 2016.

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[19] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. *CoRR*, abs/1801.04264, 2018.

[20] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos, 2019.

[21] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. ESRGAN: enhanced super-resolution generative adversarial networks. *CoRR*, abs/1809.00219, 2018.

[22] Chris M. Ward, Joshua D. Harguess, Shibin Parameswaran, and Brendan Crabb. Image quality assessment for determining efficacy and limitations of super-resolution convolutional neural network (srcnn). *Applications of Digital Image Processing XL*, Sep 2017.

[23] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2:67–78, 2014.

[24] H. R. Sheikh Z. Wang, A. C. Bovik and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[25] Roman Zeyde, Michael Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, 2010.