Semantic Prior Based Generative Adversarial Network for Video Super-Resolution

Xinyi Wu*, Alice Lucas*, Santiago Lopez-Tapia†,
Xijun Wang*, Yul Hee Kim*, Rafael Molina†, and Aggelos K. Katsaggelos*
**Dept. of Electrical Engineering and Computer Science*, *Northwestern University*, Evanston, IL, USA
†*Depto. de Ciencias de la Computación e I.A.*, *University of Granada*, Granada, Spain

***Abstract*—Semantic information is widely used in the deep learning literature to improve the performance of visual media processing. In this work, we propose a semantic prior based Generative Adversarial Network (GAN) model for video super-resolution. The model fully utilizes various texture styles from different semantic categories of video-frame patches, contributing to more accurate and efficient learning for the generator. Based on the GAN framework, we introduce the semantic prior by making use of the spatial feature transform during the learning process of the generator. The patch-wise semantic prior is extracted on the whole video frame by a semantic segmentation network. A hybrid loss function is designed to guide the learning performance. Experimental results show that our proposed model is advantageous in sharpening video frames, reducing noise and artifacts, and recovering realistic textures.**

***Index Terms*—Video Super-Resolution, Generative Adversarial Networks, Semantic Segmentation, Spatial Feature Transform, Hybrid loss function**

I. Introduction

With the increasing popularity of electronic devices such as HDTV and large-screen mobile phones, Video Super-Resolution (VSR) has gained popularity. VSR aims at producing high-resolution (HR) video sequences based on low-resolution (LR) video ones so as to improve user experience. Algorithms that tackle the super-resolution (SR) problem can be divided into two broad categories: model-based and learning-based algorithms [1]. Recent results seem to indicate that learning-based algorithms, especially those using the GAN framework [2], [3], outperform model-based methods significantly [4]–[6].

Conventional learning-based algorithms use large training databases of HR and LR video frames to learn mappings from LR to HR video sequences. Most VSR algorithms use a short LR video window around a central frame to reconstruct it, see, for instance, [1]. The main VSR strategies may be divided into three categories. The first one is based on the GAN framework, see [1], [2]. This approach uses the generator to learn the LR to HR mapping and enhances the learning performance by introducing the discriminator to judge the quality of the HR estimation. The second strategy is the combination of semantic content and SR processing, exampled in [7], [8]. Semantic information, such as object categories, texture styles and edge maps, improves the performance of SR methods. The third category uses patch-wise training for neural networks, e.g. [6], [9]. Multiple frame patches are provided as input to the networks to increase their learning efficiency. The literature shows that these three strategies have been proven very beneficial to SR methods. However, so far work merging those three strategies together into one framework has seldom been reported. This is the contribution of this work.

In this paper, we propose a semantic-prior based GAN model for VSR problems. As the name suggests, it combines GANs, semantic priors and patch-based training strategies into a single framework. We also present a detailed explanation of our approach to extract semantic information from each frame patch and combine it with the current generator. With the aid of a hybrid loss function, our proposed model produces better results than our previous work VSRRestFeatGAN, which has achieved higher performance compared to the state-of-the-art VSR models [1].

II. Methodology

The goal of a semantic image segmentation network is to separate different objects by identifying their textural features [10]. When combined with a GAN-based approach, the semantic information enhances the performance of the GAN to accurately reconstruct the HR texture. In our previously proposed VSR model VSRResFeatGAN [1], we implemented a GAN-based residual network to learn the mapping from multiple LR frames $y_{t-k}, ..., y_{t-1}, y_t, y_{t+1}, ..., y_{t+k}$, which are incorporated into the vector Y_t, to the HR central frame x_t as:

$$x_t = f(Y_t) = f(y_{t-k}, ..., y_{t-1}, y_t, y_{t+1}, ..., y_{t+k}). \quad (1)$$

When compared to the current competing state-of-the-art VSR networks, VSRResFeatGAN has been confirmed to generate estimations with higher quality [1]. However, the GAN architecture often generates artifacts, resulting in the unnatural visual perception of texture areas. To recover HR images with better perceptual quality and more realistic textures, we introduce an auxiliary semantic segmentation network to guide texture reconstruction by providing a semantic prior S for each pixel in the frame,

$$S = (P_{C_1}, P_{C_2}, ..., P_{C_n}), \quad (2)$$

This work was supported in part by the Sony 2016 Research Award Program Research Project. The work of SLT and RM was supported by the Spanish Ministry of Economy and Competitiveness through project DPI2016-77869-C2-2-R and the Visiting Scholar program at the University of Granada. SLT received financial support through the Spanish FPU program.

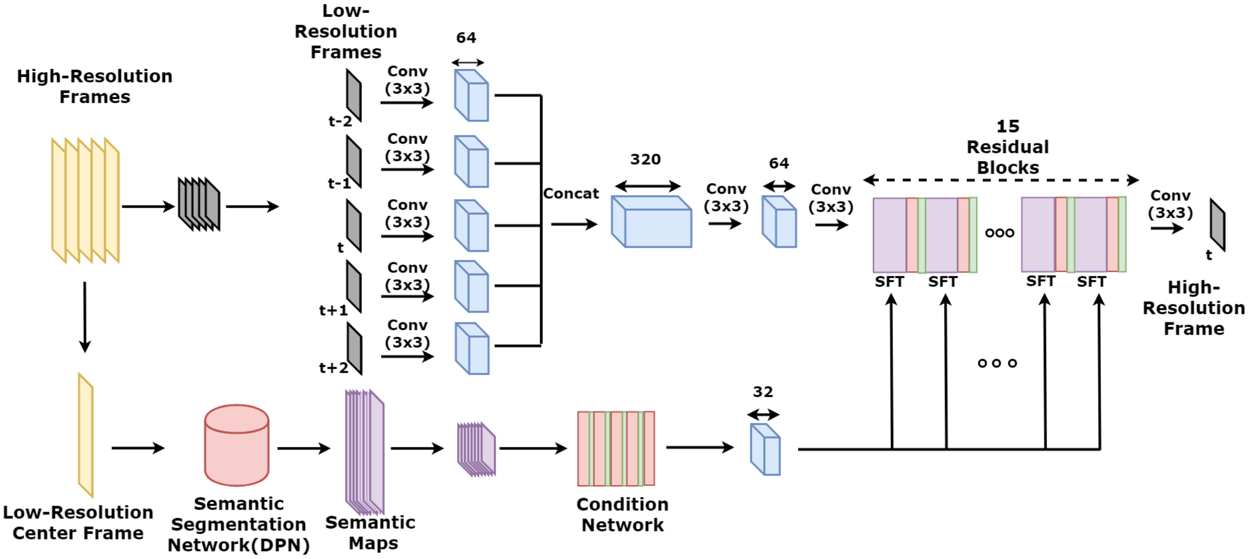

Fig. 1. The architecture of the generator for our proposed S-VSRResFeatGAN.

where P_{C_n} denotes the probability that the pixel belongs to a specific category C_n. Then (1) is reformulated as:

$$x_t = f(y_{t-k}, ..., y_{t-1}, y_t, y_{t+1}, ..., y_{t+k} jS). \qquad (3)$$

We refer to this framework as Semantic-VSRResFeatGAN (S-VSRResFeatGAN).

A. Basic Architecture

Fig. 1 shows the new framework of our generator in S-VSRResFeatGAN. Since the use of Convolutional Neural Networks (CNN) is a well established method to solve super-resolution problems, the use of patches, as short-time and stationary signals for inputs, has been found suitable for networks to converge. When training with patches, the complexity for learning also decreases due to the limited spatial extent. Therefore, with the purpose to exploit the advantages of video processing, the generator is provided with 5 bicubically upscaled LR frame patches at times $t \quad 2, t \quad 1, t, t+1, t+2$ to recover the HR frame patch t. The introduction of multiple patches from different frames strengthens the learning of high frequency patterns. Since the content of video frames is relatively stable in the time slot within 5 frames, we only make use of semantic segmentation map for the central frame patch t. The semantic information is extracted by a condition network, which consists of 5-layer convolutions followed by a Leaky Rectified Linear Unit (LeakyReLU) activation function, as shown in Fig. 2.

Feature maps of input frames as well as the semantic prior are the input to 15 residual blocks for training. We follow the design of residual blocks in [1], which are composed of two convolution layers with 3x3 kernel and ReLUs. As Wang et al. [8] found out that the spatial feature transform works better with semantic information, we incorporate it into our residual blocks to guide texture learning, see Fig. 3.

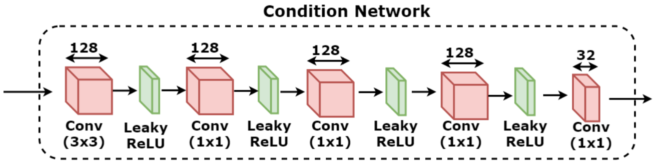

Fig. 2. The condition network for extracting semantic information from semantic maps

To apply spatial feature transforms to the current GAN-based framework, two extra convolution layers are used to transfer the semantic information into parameters A and b. These parameters are then conveniently combined with feature maps by the following affine transformation

$$SFT(F) = A \quad F \quad b, \qquad (4)$$

where F denotes the given feature maps, b and A represent the parameters learned from semantic information and   and   denote element-wise multiplication and addition, respectively [8]. With this framework, the whole generator is semantically controlled by the semantic prior to generate natural HR textures. In order to produce images of high perceptual quality, apart from the traditional mean square error loss, we include feature loss and GAN loss to constrain learning, as detailed in [1].

B. Patch-based Semantic Segmentation Prior

Our S-VSRResFeatGAN adopted the patch-size training strategy onto the previously proposed VSRResFeatGAN model [1].

However, semantic segmentation networks generally work on whole images, resulting in low performance when tested on patches. Provided with relatively little information and a small receptive field, the semantic segmentation network has difficulties in providing accurate semantic labels for each

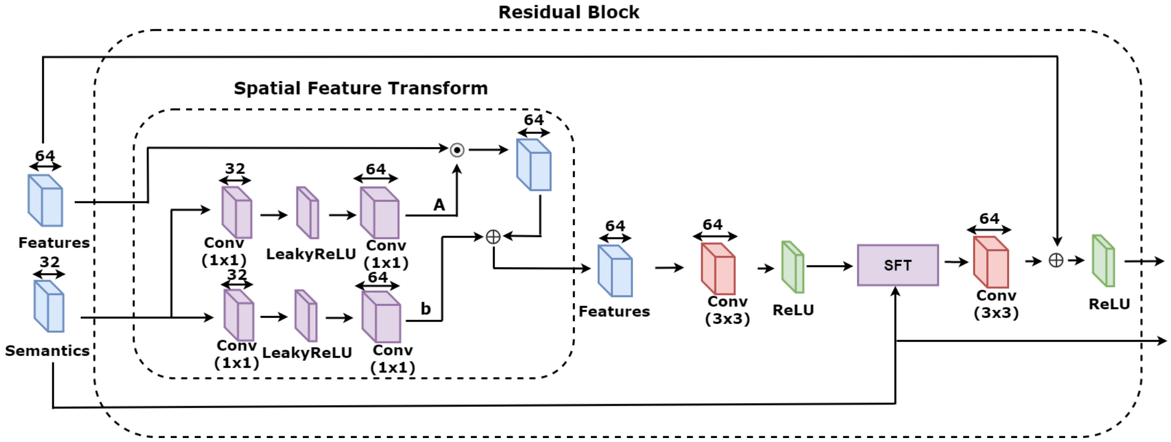

Fig. 3. The new residual block with the introduction of two spatial feature transform layers.

patch. In order to maintain the accuracy of semantic segmentation under the framework of patch-wise training, we segment whole frames and then map the global semantic map to the local space. A sliding window is utilized to capture the local semantic information from the global one. By using such patches of semantic priors, the network enhances its local semantic understanding during training, reconstructing realistic HR textures.

The network we used for semantic segmentation is borrowed from Deep Parsing Network (DPN), which has demonstrated its outstanding segmentation performance, see [11].

C. Loss Functions

In order to direct our network to generate images with better perceptual quality, multiple losses have been combined to influence the reconstruction [1], more specifically, a convex combination of pixel, feature, and adversarial losses has been used. This combined loss is designed to push our HR estimation closer to the ground truth not only in the pixel space, but also in the feature space.

To enforce the network to create accurate reconstruction in the pixel space, we make use of the Charbonnier loss:

$$\gamma(x, G_\theta(Y)) = \sum_i \sum_j \sqrt{(x_{i,j} \quad G_\theta(Y_{i,j}))^2 + \epsilon^2}, \tag{5}$$

where i, j denotes pixel location and $\epsilon > 0$, approximating the L1 regularizer [1]. However, only introducing pixel-space loss is not enough to obtain perceptually satisfying images, since the outputs result in large areas of texture being blurred. A VGG based feature loss is added to further enhance edge and texture details. With the use of the 3rd and 4th convolution layers inside the VGG network, our HR estimation and the ground truth are mapped into a learned feature space to compute the distance between predicted and ground-truth high-resolution frames. To the pixel-space loss we added $\gamma(VGG(x), VGG(G_\theta(Y)))$. With the introduction of semantic information, the VGG-based feature loss constrains the network better and produces high-quality outputs.

We use the discriminator architecture in [1]. While the generator aims to obtain super resolved images close to real ones by minimizing the function

$$L_{g(\theta)} = \mathbb{E}_Y \left[\quad \log D_\phi \left(G_\theta \left(Y \right) \right) \right], \tag{6}$$

the discriminator is trained to distinguish between real and generated HR images through maximizing the loss

$$L_{d(\phi)} = \mathbb{E}_x \left[\log D_\phi \left(x \right) \right] + \mathbb{E}_Y \left[\log \left(1 \quad D_\phi \left(G_\theta \left(Y \right) \right) \right) \right]. \tag{7}$$

With such adversarial training [12], we push the network to provide HR frames of high perceptual quality.

To summarize, combining all the loss functions above, our final loss for the generator is formulated as:

$$\begin{aligned} L_{total(\theta)} = \alpha \sum_{(\mathrm{Y},\mathrm{x}) \in T} & \gamma(VGG(x), VGG(G_\theta(Y))) \\ & + \beta \mathbb{E}_Y \left[\quad \log D_\phi \left(G_\theta \left(Y \right) \right) \right] \\ & + (1 \quad \alpha \quad \beta) \sum_{(\mathrm{Y},\mathrm{x}) \in T} \gamma(x, G_\theta(Y)), \end{aligned} \tag{8}$$

where α and β are non-negative values with $\alpha + \beta < 1$ and are determined experimentally to control the weight of each loss component. The loss for our discriminator is the same as in (7).

III. Experiments

A. Training and Parameters

Our experiments were conducted on the publicly available 4K Myanmar video dataset [13]. We sampled frames from the video sequences and downsampled them to 960x540 pixels to reduce memory requirements. Our semantic segmentation model, which is borrowed from [8], [11], was pre-trained on the MS-COCO dataset [14] and fine-tuned with the ADE20K dataset [15]. Images will be forced to produce segmentation maps with 8 categories (buildings, plants, sky, mountains, water, animals, grasses and background), which correspond to object instances frequently seen in the outdoor scenes in our training dataset.

Thus, for each HR color frame at time t, we extracted an LR color frame and fed it to the semantic segmentation network to generate a set of semantic maps with the corresponding probabilities of each category, with size $1 \quad 8 \quad H \quad W$. To synthesize our 2D patch-wise training dataset, we split the luminance channel y from the HR frame by Matlab's *rgb2ycbcr* function and performed patching. Each sample in the training input consists of 5 36x36 LR grey patches at times $t \quad 2, t \quad 1, t, t+1, t+2$ and the corresponding 8 channels of the 36x36 semantic prior for time t. Meanwhile, the $36 \quad 36$ HR grey patch at time t is defined to be the ground truth used during training. All the LR objects are obtained by MATLAB's *imresize()* to bring the low-resolution patches to the same spatial extent as the high-resolution ones. In our experiment, we use a total of 910,000 patch pairs for training. Our network processes only the luminance channel y, which is then combined with the bicubically upscaled cb, cr channels to create the final color output.

In order to facilitate the stability of the training process for the semantic GAN, we first followed the pretraining procedure detailed in [1], where a model is first trained with the MSE loss in pixel-space only before using it as initial weights in the combined loss training. The authors of [1] explain that this is necessary for large scale factors to avoid subsequent failure of the GAN-based training. Thus, we obtained the pretrained VSRResFeatGAN model provided by [1]. We note that this model does not include the spatial feature transform layers. Therefore, in the subsequent GAN training with semantic prior, apart from the weights of the spatial feature transform layers, the remaining parts in the generator of our S-VSRResFeatGAN were initialized with the weights transferred from the pretrained pixel-wise VSRResFeatGAN model [1]. For the spatial feature transform layers, we use the initialization in Kaiming et al. [16]. The discriminator is trained from scratch. The combined loss in (8) was used to control the semantic GAN network to produce estimation of high perceptual quality. We used the same hyper-training parameters and loss contributions as those specified in [1]. In these settings, we find that our network converged after the training for 30 epochs.

B. Evaluation Results

Our model was trained for 2, 3, and 4 SR factors. In order to test the performance of our model on a general dataset, we test our model and compare it with VSRResFeatGAN on the VidSet4 dataset [17], which is a commonly used video dataset for the assessment of video SR models.

Recently, traditional image quality metrics such as PSNR and SSIM were found not to accurately estimate the perceptual quality of an HR estimation. For the purpose of accurately reflecting the behavior of our model on perception, when testing VSRResFeatGAN [1], we use the novel criterion proposed in Zhang et al. [18] to measure the perceptual similarity between two images. In the experiments for VSRResFeatGAN [1], the perceptual similarity network shows an outstanding human perceptual judgment for determining the sharpness of video frames. Therefore, we keep this metric to assess the texture reconstruction of our S-VSRResFeatGAN.

We use PSNR, SSIM and Perceptual Distance (PercepDist) to compare our S-VSRResFeatGAN VSR model to the current state-of-the-art VSRResFeatGAN model [1]. Note that the segmentation accuracy significantly affects the behavior of our model, which means that a better segmentation contributes to better outputs. Since our semantic segmentation network is trained for some special scenes, it is necessary to make sure that our S-VSRResFeatGAN takes advantage of the right semantic information when testing on the VidSet4 dataset [17]. Within the four sequences in VidSet4, we particularly focus on the detailed results in the Foliage sequences, which provides relatively accurate semantic prior in testing. The results are shown in tables I and II.

TABLE I
AVERAGE PERFORMANCE COMPARISON ON VIDSET4 DATASET

Scale Factor	VSRResFeatGAN PSNR/SSIM/PercepDist	S-VSRResFeatGAN PSNR/SSIM/PercepDist
u2	30.90/0.9241/0.0283	**31.19/0.9316/0.0269**
u3	26.53/0.8148/0.0668	**26.79/0.8238/0.0659**
u4	24.50/0.7023/**0.1043**	**24.81/0.7146**/0.1086

For the PercepDist metric, smaller is better.

TABLE II
QUANTITATIVE COMPARISON ON FOLIAGE SEQUENCES

Scale Factor	VSRResFeatGAN PSNR/SSIM/PercepDist	S-VSRResFeatGAN PSNR/SSIM/PercepDist
u2	29.71/0.9108/0.0342	**30.48/0.9237/0.0258**
u3	25.29/0.7544/0.0736	**25.80/0.7734/0.0666**
u4	23.20/0.5974/0.1203	**23.99/0.6421/0.1146**

Table I suggests that our S-VSRResFeatGAN model outperforms VSRResFeatGAN on almost all the metrics on the VidSet4 dataset [17]. Furthermore, Table II shows that in terms of the PercepDist metric, our model outperforms the VSRResFeatGAN model by a larger margin when provided with a relatively accurate segmentation. We provide qualitative examples, in Fig. 4 and 5. Focusing on the zoomed-in areas in the frames we see that our S-VSRResFeatGAN reconstructs textures with less noise and the estimated textures keep shapes closer to those of the ground truth.

As mentioned before, the spatial feature transform layers in our S-VSRResFeatGAN model utilize high-level semantic information to learn texture styles. Our experiments found that it is crucial that similar types of texture styles in the testing set are also included in the training set. Otherwise, even though the segmentation is accurate, the HR estimation may have artifacts and distortions from emphasizing wrong texture styles. When the scale factor becomes larger with the sharp decrease of the informative prior, the artifacts and distortions will be more prevalent, as shown in the results of Table I. The average improvement of PercepDist gradually decreases