Joint Caption Detection and Inpainting Using Generative Network

Vismay Patel and Anubha Pandey

1 Introduction and Related Work

The task of Video Decaptioning can be simplified into solving two tasks, caption detection and filling missing information instead of the caption. There has been a lot of literature on image/video inpainting where the goal is to fill the missing patches with semantically meaningful information that is also coherent with the supporting pixels. The traditional non-learning based approaches [3, 5, 7, 8] towards image/video inpainting try to copy the information from the neighboring spatio-temporal patches which are most similar to the partially filled patch. Recent advances in deep learning use generative CNN architectures to fill the missing patch texture with generated data such as [2, 6, 9]. There has been a lot of work on text detection from images. However, it is still a challenging task to do fast and accurate automatic text removal and inpainting in video sequences.

2 Frame Level Inpainting and Caption Detection

We use an encoder-decoder based CNN model to generate the inpainted frames and the caption detection masks. The network has two branches each for the image generation and the mask generation tasks and these branches share the parameters up to first three convolution layers and the layers thereafter, are trained independently. Doing the caption detection jointly with the image generation network allows us to reuse initial few layers of the network and hence improves the efficiency of out

V. Patel · A. Pandey (✉)
Indian Institute of Technology Madras, Chennai, India
e-mail: cs16s023@cse.iitm.ac.in

© Springer Nature Switzerland AG 2019

S. Escalera et al. (eds.), *Inpainting and Denoising Challenges*,
The Springer Series on Challenges in Machine Learning,
https://doi.org/10.1007/978-3-030-25614-2_7

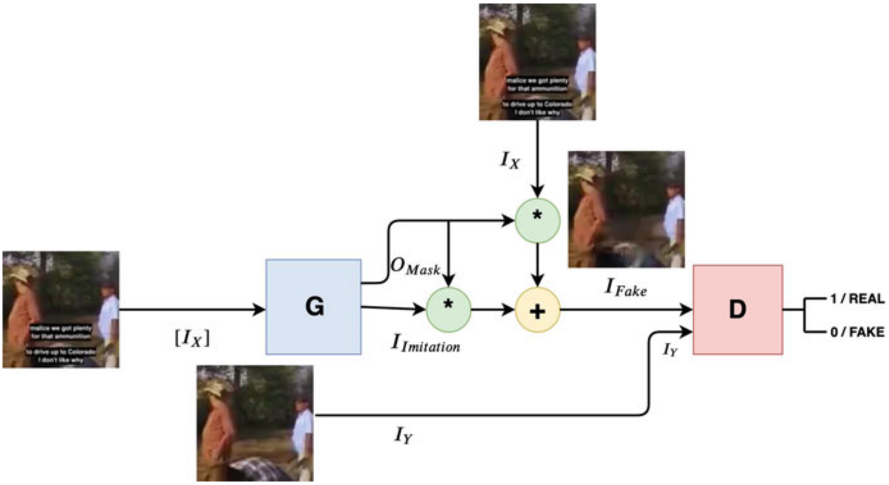


Fig. 1 Full pipeline of the Image Inpainting network. '*' represents pixel-wise multiplication of two images and '+' represents pixel-wise addition of two images

solution. The detected caption mask allows us to copy the non-masked region pixels from the original frame and masked pixels from the inpainted frame. We use a combination of Reconstruction loss, Perceptual loss, and Adversarial loss and train our model using Adam Optimization Algorithm.

Inputs to the network are frames from the captioned videos and the caption masks extracted for the corresponding frames. The masks are extracted by taking the difference between the corresponding frames of the ground truth decaptioned videos and the input videos to be decaptioned. The mask contains ones in places of captions and zeros everywhere else (Fig. 1).

2.1 Network Architecture

Our network has the following main modules:

1. `Generator Module`: This module is used to generate inpainted images and caption masks corresponding to the input video frames. It is an encoder-decoder based CNN model. In the encoder part initially, we use a series of convolution layers. We use dilated convolutions in the later stages of the encoder. First, three convolution layers are shared between both the image and mask generation tasks and later layers are divided into two branches and are trained independently. The decoder for both the branches use a series of deconvolution followed by convolution layers. We also add skip connections from the encoder module to the decoder module in the image generation branch. The skip-connections allow us to pass fine details to the coarse layers in order to generate details in the images.

The decoder tries to generate an image closer to the ground truth frame and a mask containing ones at the pixels that are captions and zeros everywhere else. The input to the Generator module is $128 * 128 * 3$ sized tensor corresponding to the frame of the input video to be inpainted. The generator architecture is shown in Fig. 2. Our network is inspired by the work of [2].

2. `Discriminator Module`: This module is used while applying GAN [1] loss to the generator. It helps in distinguishing between the real ground truth frames and the fake inpainted images from the generator. The architecture of the module is shown in Fig. 3. The discriminator outputs probability of the image being real. Thus the output value should be close to 0 for fake images and it should be close to 1 for real ground truth images.

2.2 Training

We train our network using Adam Optimizer with learning rate 0.006 and batch size 20. For first 8 epochs we train only the generator module of the network minimizing only the reconstruction loss and perceptual loss and for the next 12 epochs, we train the entire GAN network end to end on minimizing all three losses. Following loss functions have been used to train the network:

1. `Reconstruction Loss`: To generate images similar to the ground truth image, we try to reduce absolute error between generated images and the ground truth images. Similarly, in order to do caption detection, we try to reduce the squared error between the generated mask and the ground truth mask.

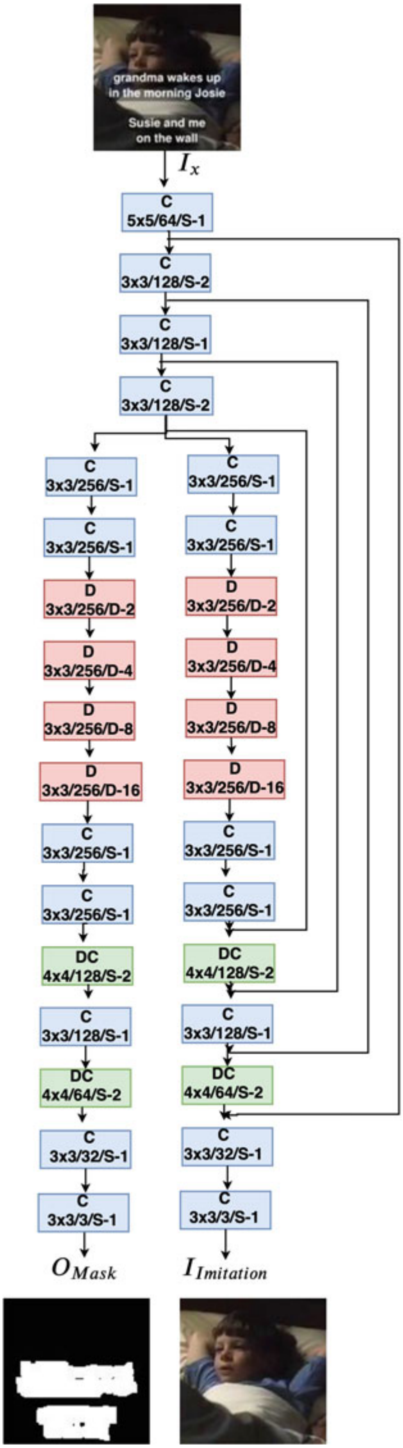
$$L_r = \frac{1}{K}\sum_{i=1}^{K}|I^i_{groundtruth} - O^i_{imitation}| + \alpha * \frac{1}{K}\sum_{i=1}^{K}(I^i_{mask} - O^i_{mask})^2 \qquad (1)$$

where, K is the batch size and α is hyper-parameter acts as a trade-off between two terms in the reconstruction loss. We found the best α value is $1 * 10^{-6}$. The α value is low in order to reduce overfitting of the cation detection branch as caption detection is easier task compared to the image generation task.

2. `Adversarial Loss`: In order to maximize the probabilities for real images and minimize the probabilities for fake images, the total discriminator loss L_d is a combination of two partial losses.

$$\begin{aligned} L_{real} &= -log(\ D(\ I_{groundtruth}\ )\ )\ ,\ \ L_{fake} = -log(\ 1\ -\ log(\ D(\ G(\ I_{input}\ )\ )\ ) \\ L_d &= L_{real} + \beta * L_{fake} \\ L_g &= -L_d \end{aligned} \qquad (2)$$

Fig. 2 Architecture of the generator module of the inpainting network. Each building block is described in Fig. 4



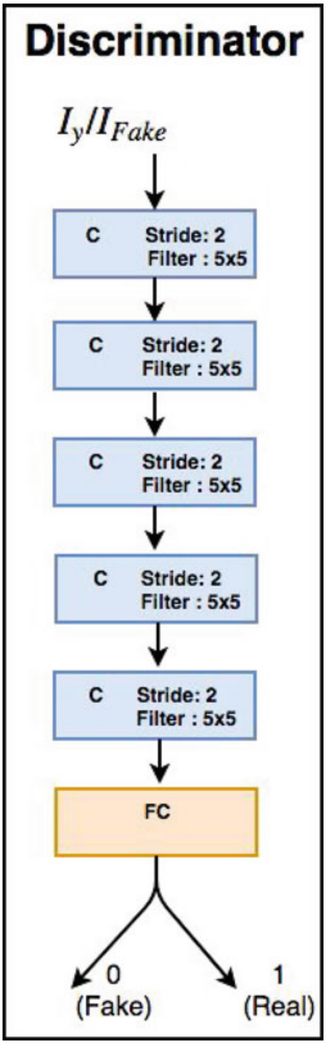


Fig. 3 Architecture of the discriminator module of the inpainting network. Each building block is described in Fig. 4

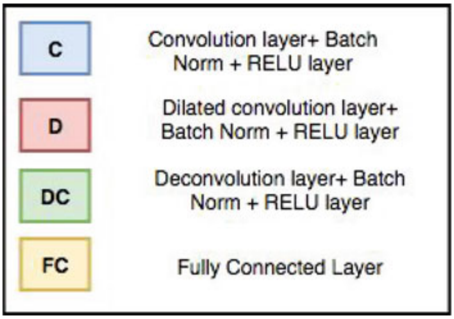

C Convolution layer+ Batch Norm + RELU layer
D Dilated convolution layer+ Batch Norm + RELU layer
DC Deconvolution layer+ Batch Norm + RELU layer
FC Fully Connected Layer

Fig. 4 Building blocks of the network. Each block in the diagram also contains the filter size/number of output channels and stride (S) or dilation (D) of the respective layer

where, D is the discriminator function and G is the generator function and β is trade-off parameter between L_{real} and L_{fake}. The best value of β is $1 * 10^{-2}$ which is decided by cross validation.

3. `Perceptual Loss` [4]: The knowledge of contextual information is very crucial for filling the correct missing pixels in any image, it helps to produce a perceptually similar image from the original image. We use pre-trained vgg network to find the perceptual difference between ground truth images and the generated images.

$$L_p = \frac{1}{K}\sum_{i=1}^{K}(\phi(I_y) - \phi(I_{imitation}))^2 \qquad (3)$$

where, ϕ represents features from pretrained VGG16 network.

3 Experiments and Results

We evaluate our model on the dataset provided in the ECCV'18 Satellite Workshop `Chalearn LAP Inpainting Competition Track 2 - Video decaptioning`. To evaluate the quality of the reconstruction, MSE, PSNR and DSSIM metrics as mentioned on the competition's website[1] are used for pairwise frame comparison.

The proposed model is implemented using tensorflow-gpu 1.6.0 framework on the top of python 3.6.4 and the platform used is Ubuntu 14.04. The training of the model takes 2.5 h for one epoch on a GeForce GTX 1080 graphics card. There are 31,686,529 parameters to be learned in the network. We trained the network on the above-mentioned dataset for 2 days up to 20 epochs.

With our proposed solution we secured third position in the competition. Performance comparison of our proposed model with that of other teams are shown in Table 1. The qualitative results of our proposed solution is shown in Table 2.

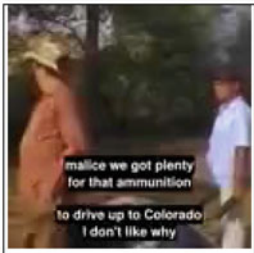
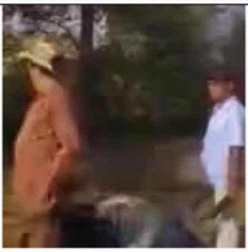
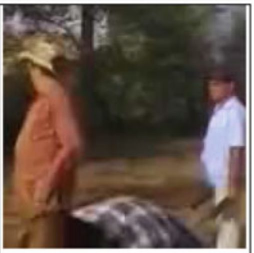
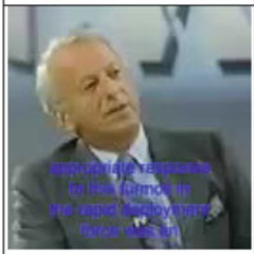
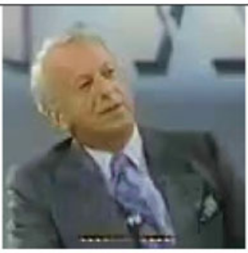
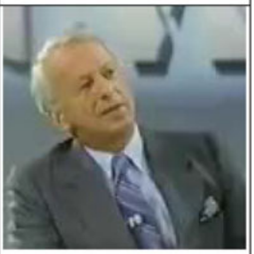
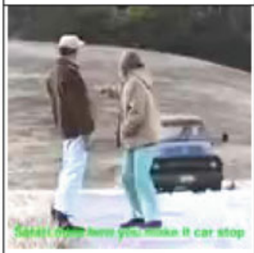
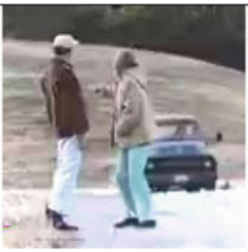
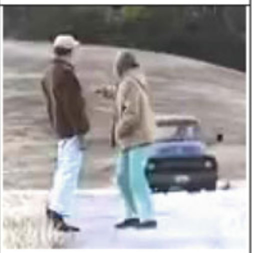
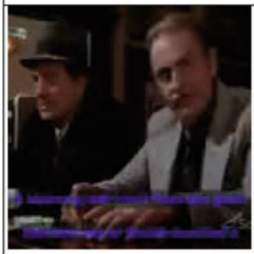
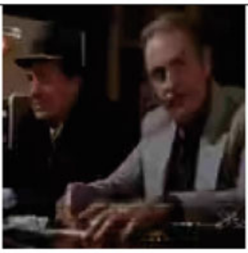
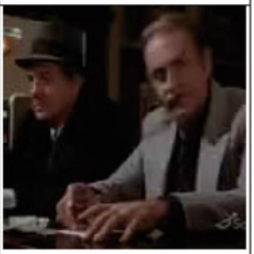
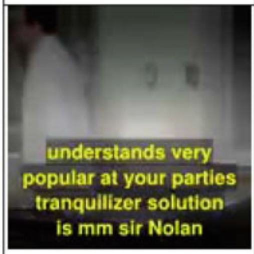
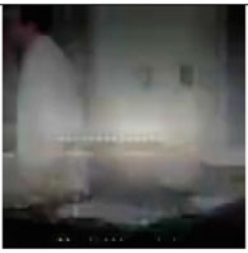
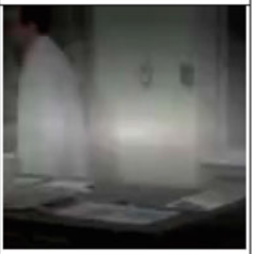
Table 1 Comparison of performance with other teams

Team name	Rank	MSE	PSNR	DSSIM
SanghyunWoo	1	0.0011	33.3527	0.0404
hcilab	2	0.0012	33.0228	0.0424
anubhap93	3	0.0012	32.0021	0.0499

We are team anubhap93

[1] https://competitions.codalab.org/competitions/18421#learn_the_details-evaluation.

Table 2 Qualitative results

Input Frame	Inpainted Frame	Ground Truth
		
		
		
		
		

4 Conclusions

In this work, we have proposed an end-to-end solution for de-captioning of diverse video clips having text overlays of different size, location, background, and color. The network can simultaneously do frame level caption detection and inpainting. However, this method requires individual frames from the clip to do its task which lacks the temporal context required to produce the desired result.

In future work, we aim to improve performance by exploiting the temporal information of the video clips. Techniques used in intermediate frame prediction can be employed to make the network temporally-aware. We aim to explore models that use both temporal and semantic information.

References

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems, pp. 2672–2680 (2014)
2. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Transactions on Graphics (TOG) **36**(4), 107 (2017)
3. Jia, Y.T., Hu, S.M., Martin, R.R.: Video completion using tracking and fragment merging. The Visual Computer **21**(8–10), 601–610 (2005)
4. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision, pp. 694–711. Springer (2016)
5. Newson, A., Almansa, A., Fradet, M., Gousseau, Y., Pérez, P.: Video inpainting of complex scenes. SIAM Journal on Imaging Sciences **7**(4), 1993–2019 (2014)
6. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2536–2544 (2016)
7. Shen, Y., Lu, F., Cao, X., Foroosh, H.: Video completion for perspective camera under constrained motion. In: null, pp. 63–66. IEEE (2006)
8. Wexler, Y., Shechtman, E., Irani, M.: Space-time video completion. In: null, pp. 120–127. IEEE (2004)
9. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5505–5514 (2018)