Relevance of Sparse pixel representations to Lossy Video Compression

Rahul Nittala
Saarland University
rani00002@stud.uni-saarland.de

Raj Mohan Tumarada
Saarland University
ratu00001@stud.uni-saarland.de

Ayushi Churiwala
Saarland University
aych00001@stud.uni-saarland.de

Abstract

Classical video compression techniques improve the coding efficiency by reusing pixels from previously decoded frames via motion and residual compensation. Recently, GAN-based deep models were applied to images for sparse pixel-based representation. In this project, we explore how extendable these sparse pixel selection techniques employed for image compression are to the task of lossy video compression. Wasserstein GANs with recurrent GRU units are used to reliably learn the relevant pixels to select across frames and also the natural distribution of the missing regions as a function of the selected pixels. To ensure superior video reconstruction quality both temporal and spatial discriminators are used. We show qualitative and quantitative results on the UCF-101 dataset and discuss the strengths and shortcomings for the discussed method.

1. Introduction

In this digital age images, audio and videos constitute the primary methods of information dissemination. We enjoy and prefer high-resolution images, videos, and high-bit rate quality audio. However, in situations where bandwidth limitations exist or media transfer latency needs to be minimized, compression techniques become important(even at the expense of some loss). Due to the large size of videos, lossy compression becomes more relevant than other media forms. Compression helps reduce file size, allows shorter uploads, lowers hosting costs ensures fewer bandwidth requirements during playback.

Image inpainting is a reconstruction process where given some pixels in an image, the task is to fill the missing parts of it. An extension to this is the spatial optimization problem. It is formulated as: select a fraction of known pixels such that the image can be impainted to a good quality. [8]. This selection of important/sparse pixels can have applications in inpainting-based compression [5] [9], adaptive sampling [3].

Neural networks are recently being used to solve the spatial optimization problem [3] [1]. The process is split into two stages: mask generation(for pixel selection) and inpainting(for reconstruction). Jointly training them has been shown to work well not only quantitatively but also in identifying good pixels with a certain selection density that tend to concentrate around high frequent regions in the image. Therefore, these sparse pixels can be viewed as a way to achieve lossy image compression. As shown in Fig.1, with only 5% of the total pixels available, methods were proposed that would return a good quality reconstruction of the original images. Since a video is a series of a few still images (frames) on which inpainting can be applied, we believe that a sparse pixels based representations can be advantageous towards video compression task.

The organization of the report is as follows: In section[2] we briefly discuss the prior work which was done in to solve the spatial optimization problem. Section[3] talks about the proposed approach and the architectural details of the models employed. We then see to what extent we solve the video compression task from a qualitative and quantitative standpoint in section[4]. We also comment on the relevance and viability.

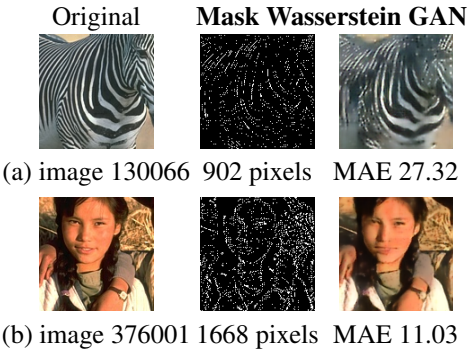


Figure 1. Pixel selection and Inpainting for Mask generation + WGAN method. *Image Source: A Wasserstein GAN for Joint Training of Inpainting and its Spatial Optimisation [8]*

2. Related Work

To our best knowledge, image inpainting networks using WGAN have not been used for video compression. The

network of Alt et al. [1] optimizes masks using a non-binary confidence map for homogeneous diffusion inpainting (not deep inpainting). Dai et al. [3] first trained an inpainting network NetE with randomly generated masks. With the NetE now frozen, a mask network NetM was trained. The mask generation and inpainting processes are inherently strongly coupled. But due to the use of masks that are non-binary the joint training did not yield satisfying results. [8] applies quantization technique to achieve binary masks during training.

One of the most successful stochastic methods for sparse pixel representation is Probabilistic Sparsification & Nonlocal pixel exchange(PS+NLPE) [7]. Though PS+NLPE produces great results, it has a very long processing time. WGANS [2] were shown to be very successful for inpainting [12]. The most recent method MG+WGAN [8] proposes a joint training mechanism that encourages strong coupling between mask and image inpainting networks.

Existing video compression techniques using deep learning majorly use tucker tensor decomposition [4] or dictionary learning algorithms [6]. However, for our approach, we extend the architecture of MG+WGAN [8].

3. Methods

Our methodology displayed in Fig.2 resembles MG+WGAN [8] as the skeleton architecture. We apply a GRU on top of the generators as well as on temporal discriminator. This addition helps improve the mask generator and inpainting as we no longer perform inpainting for an image but over a set of frames for a video. The mask generator **m** maps the original video (sequence of image frames) $f_i \in \mathrm{R}_i^N$ where i is the i^{th} frame in the video and random seed $r_i \in \mathrm{R}^N$ to a binary mask $\mathrm{b}_i = \mathbf{m}(\mathrm{r}_i, \mathbf{f_i})$. The generator **g** will use this mask superimposed with the base frame $\mathbf{b_i} \odot \mathbf{f_i}$ as input to generate an inpainted frame u_i from different random seed.

We use two discriminators – a Temporal Discriminator, D_t and a Spatial Discriminator D_s. D_t judges any temporal discrepancies across the length of the video and D_s judges any high-resolution details. Instead of using the discriminator frame-wise, we task it to distinguish true video distribution from the generated one as a whole. At the final frame we get a score d(u) which tells us the similarity of the decompressed video to the real one. We used Spectral Norm on discriminator weights to satisfy Lipschitz property which forms the base for Wasserstein distance. The mask generator, with a deensity loss is used that measures the deviation of the percentage of known total pixels with the mask's target density **D** is combined to the Wasserstein loss of the generator and discriminator since mask **m** influences the inpainted frame **u**. This mask density loss along with the video reconstruction loss i.e. the MSE loss between the actual and generated video is given as:

$$E_{f\sim P_t, r\sim P_s}(\|(\Sigma M)/N - D\|_2 + \beta \|f - u\|_2)$$

where **N** is the Total number of pixels and β is the Relative weighting factor. The generator takes a sample from a source distribution P_s along with the mask and maps it to a representative of P_t.

Our base model has four networks: a mask generator, frame generator, a temporal discriminator, and a spatial discriminator. Inspired by the architecture proposed by Pascal Peter [8] an hourglass structure is used for the frame generator which successively subsamples the input data and upsamples it again to get the output. Skip connections between the corresponding convolution and transposed convolution blocks ensure a natural flow of relevant information from previous layers to help avoid the information bottleneck. Downsampling is performed by Conv blocks with convolutions of filter size 5 × 5 and dilation rates 0, 2, and 5. Their outputs are appended, and ELUs(exponential Linear Units) are used as activations. The transposed convolution of the last layer has hard sigmoid activation to restrict the image domain to lie in the range[0, 1]. Both the discriminators follow a basic Funnel blocks like architecture for downsampling, where each Block has 5 × 5 convolutions of stride 2 with Leaky ReLU activations.

In order to overcome the drawbacks of using a non-binary mask, we use the Binarization block as suggested in [8]. A rounding function($b(c) = \lfloor c+0.5 \rfloor$) can be used to emulate the binary nature of the masks. But such a formulation is not differentiable and therefore, doesn't fit in an end-to-end neural network training scenario. Theis et al. [11] proposes using a pseudo gradient for rounding function. They approximate the rounding function as $\lfloor c + 0.5 \rfloor \approx c$. The gradient would then be an identity. No processing is required and the gradients received by the Binarization block can be sent back as is.

4. Experiments

4.1. Experimental Setup

We have trained the **full model**, as shown in Fig.2, on the *ApplyEyeMakeup* class of the UCF-101 dataset [10] . For training, each video in the training dataset is divided into multiple clips with 20 frames per clip. Each frame is center clipped to a resolution of 128×128. In addition to the full model, we wanted to see how important the mask generator network is in terms of performance gain as it increases the time and computational overhead during training as well as validation stages. We trained two models that directly use masks with 5%(**No Mask Gen(5%)**) and 10%(**No Mask Gen(10%)**) of the total pixels selected instead of using the mask generator network. These masks are randomly sampled from a uniform distribution.

We were only able to train each of the above mentioned models for 15 epochs(more details in 4.4). We used Adam

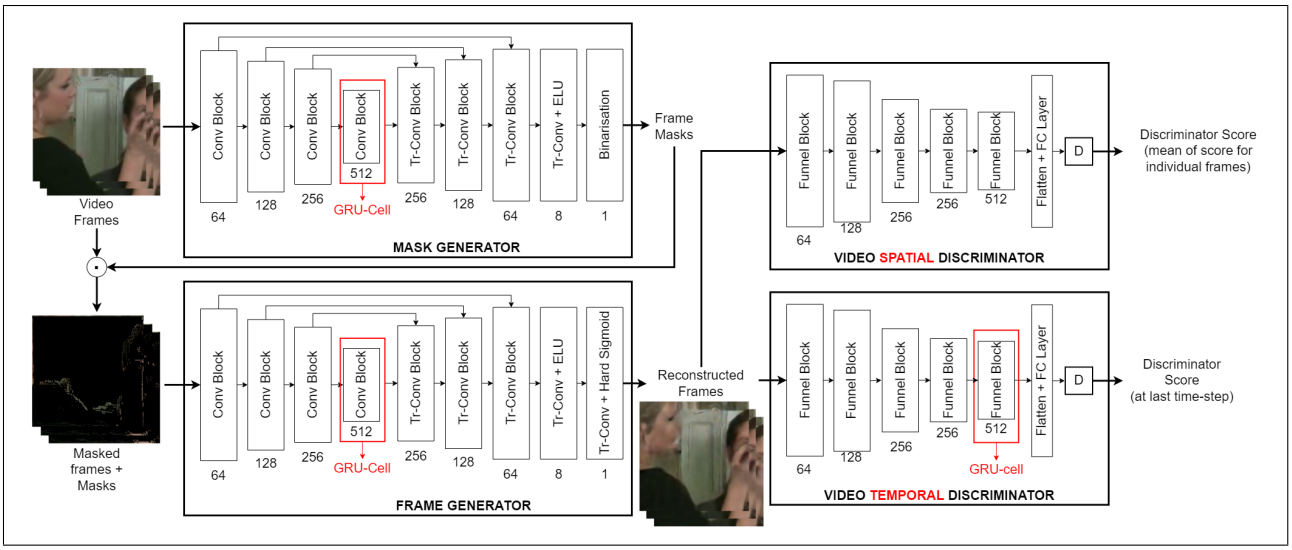


Figure 2. Video Compression: Proposed architecture.The arrows denote forward passes. Conv Blocks and Tr-Conv Blocks denote convolutional and transposed convolutional blocks.

optimizer and constant learning rate scheduler which decays the learning rate by 0.9 for every 10000 steps. The batch size was set to 2. The learning rates of generators and discriminators were set to $5e-5$ and $1e-5$ respectively.

4.2. Quantitative results

To show the quantitative performance of the models we used PSNR(peak signal to noise ratio) and SSIM(Structural similarity index measure) which are the commonly used metrics for testing reconstruction quality. The results are shown in the table[1]. The No Mask Gen cases vastly outperform the Full model case. It is important to note though that all the models were not trained to convergence and GANs typically need a lot of epochs to converge. Further the Full model case has additional mask generator network parameters which also needed optimization. These might explain the poor performance of the full model. We will look at why the models weren't trained to convergence in the section 4.4.

Model	PSNR	SSIM	Selected pixels%
Full Model	17.6	0.58	5.6%
No Mask Gen(5%)	22.67	0.72	5%
No Mask Gen(10%)	25.74	0.75	10%

Table 1. Quantitative perfromance. **No mask generator** cases perform better when models are trained only for 15 epochs.

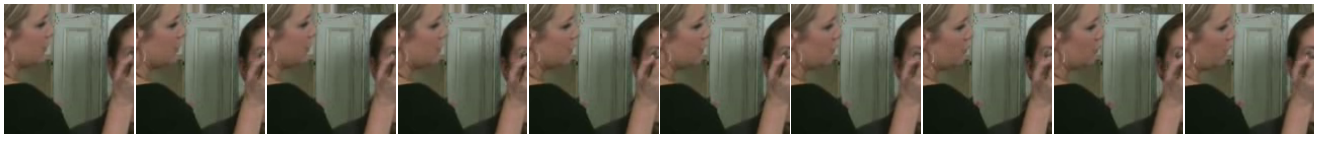
4.3. Qualitative results

Fig.[3] shows the masked video and its reconstruction for the first 10 frames of a test video. Looking at the masked video we can see that the network captures the relevant edges present in each frame. It then uses this information to inpaint the missing regions. But The person on the left doesn't seem to be reconstructed properly. We think training the model until convergence can alleviate this problem. And using either Bi-directional GRU units or making more units other than just the bottleneck layer in the model recursive might further improve the performance. The reconstructions for No mask gen 5% and 10% models are shown in the appendix section[6]

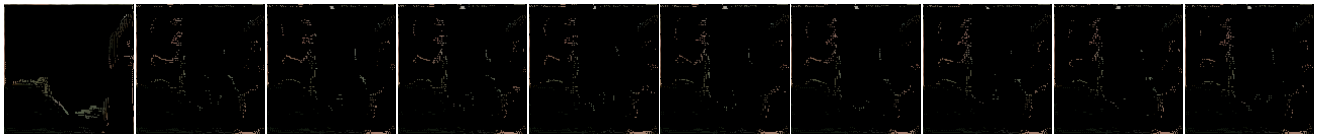
We also tested the model on a test clip from the *Archery* class which was no part of the training dataset. The results are shown in Fig.[4]. We can clearly see that the masked video generated is not of a good quality where the entirety of the first frame is chosen and the latter ones are mostly blank. The reconstructed video replaces the blank regions with the color of skin. Human faces were a major part of *ApplyEyeMakeup* videos. This means that the model learnt is not very generalizable. Hence, to create a general purpose video compressor, we need to train the model on a multitude of video distributions. On the contrary, though not great, we observed that the 5% and the 10% pixels gave more visually pleasing results. Look at the appendix section[6] for the results.

4.4. Training issues

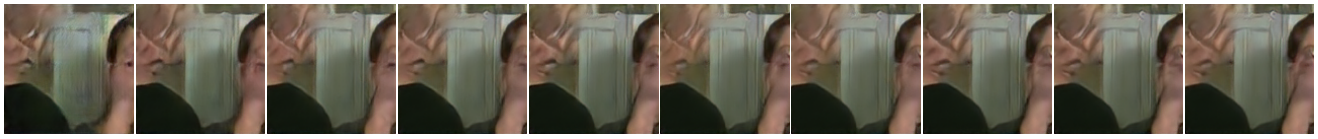
During training we were only able to load 2 videos of 20 frames per a single batch to train the model on a 16GB GPU RAM. It also took us a day to train 2.5 full epochs. Training on more distributions or the use of more structural sophistication(more GRU units/ bidirectional units) would only exacerbate the already existing space and time com-



(a) Test video clip



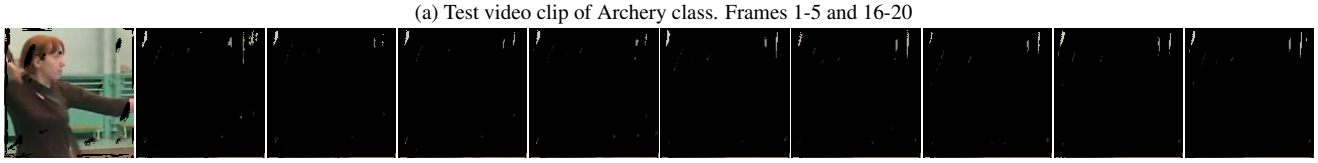
(b) Masked video generated by Mask Generator of the full model



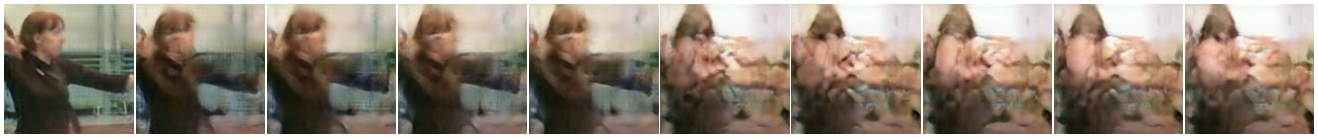
(c) Reconstructed video generated by the Frame Generator of the full model

Figure 3. Qualitative performance of the full model. Results shown are the first 10 frames of a test clip from *ApplyEyeMakeup* class

(a) Test video clip of Archery class. Frames 1-5 and 16-20



(b) Masked video generated by Mask Generator of the full model. Frames 1-5 and 16-20



(c) Reconstructed video generated by the Frame Generator of the full model. Frames 1-5 and 16-20

Figure 4. Out of distrubtion performance of the full model. Results shown are of a test clip from *Archery* class. The frames shown are first 5 frames followed by the last 5 of a 20 frame clip.

plexity. Further the deep models trained would be applicable only to a particular video resolution and compression quality. We need to train more models to accommodate different video resolutions and compression rates.

4.5. Compressed file size

For the experimental setup that we used, choosing 5% of the total pixels for 20 frames would be equivalent to saving one frame for every 20 frames. We additionally need to save the selected pixel locations(masks). We need to devise efficient containers and algorithms to save both.

5. Conclusion

We have seen a recurrent GAN-based approach to Video compression. For the training that was done(15 epochs) performance of the full model is not very encouraging when compared to random mask selection scenario. This is subject to change when trained until convergence. The widely used video codecs like HEVC easily outperform the models studied in this report. In addition to the poor performance, the studied models are not generally applicable across different video distributions or resolutions or compression rates.

References

[1] T. Alt, J. Weickert, , and P. Peter. Learning sparse masks for diffusion-based image inpainting. *arXiv:2110.02636[eess.IV]*, 2021. 1, 2

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 2

[3] Qiqin Dai, Henry Chopp, Emeline Pouyet, Oliver Cossairt, Marc Walton, and Aggelos K. Katsaggelos. Adaptive image sampling using deep learning and its application on x-ray fluorescence image reconstruction. *IEEE Transactions on Multimedia*, 22(10):2564–2578, 2020. 1, 2

[4] Samiran Das. Hyperspectral image, video compression using sparse tuckertensor decomposition. 2021. 2

[5] I. Galic, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, and H.-P. Seidel. Image compression with anisotropic diffusion. *Journal of Mathematical Imaging and Vision, vol. 31, no. 2–3, pp. 255–269*, 2008. 1

[6] Maziar Irannejad and Homayoun Mahdavi-Nasab. Block matching video compression based on sparse representation and dictionary learning. 2018. 2

[7] Markus Mainberger, Sebastian Hoffmann, Joachim Weickert, Ching Hoo Tang, Daniel Johannsen, Frank Neumann, and Benjamin Doerr. Optimising spatial and tonal data for homogeneous diffusion inpainting. In Alfred M. Bruckstein, Bart M. ter Haar Romeny, Alexander M. Bronstein, and Michael M. Bronstein, editors, *Scale Space and Variational Methods in Computer Vision*, pages 26–37, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 2

[8] Pascal Peter. A wasserstein gan for joint learning of inpainting and its spatial optimisation. *arXiv preprint arXiv:2202.05623*, 2022. 1, 2

[9] C. Schmaltz, P. Peter, M. Mainberger, F. Ebel, J. Weickert, and A. Bruhn. Understanding, optimising, and extending data compression with anisotropic diffusion. *International Journal of Computer Vision, vol. 108, no. 3, pp. 222–240*, 2014. 1

[10] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2

[11] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017. 2

[12] Daniel Vašata, Tomáš Halama, and Magda Friedjungová. Image inpainting using wasserstein generative adversarial imputation network. In *International Conference on Artificial Neural Networks*, pages 575–586. Springer, 2021. 2

6. Appendix

Please look at the following page for furtherresults:

Figure 5. Test Video clip from ApplyEyeMakeup class.

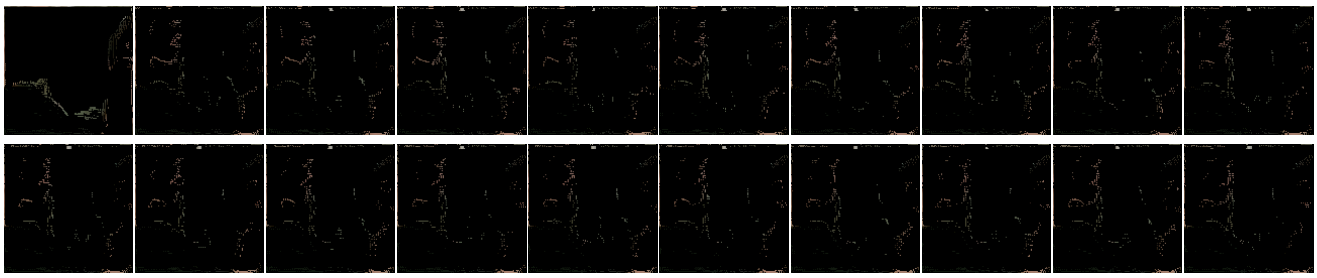


Figure 6. Masked Video generated by Mask Generator of the full model from ApplyEyeMakeup class.

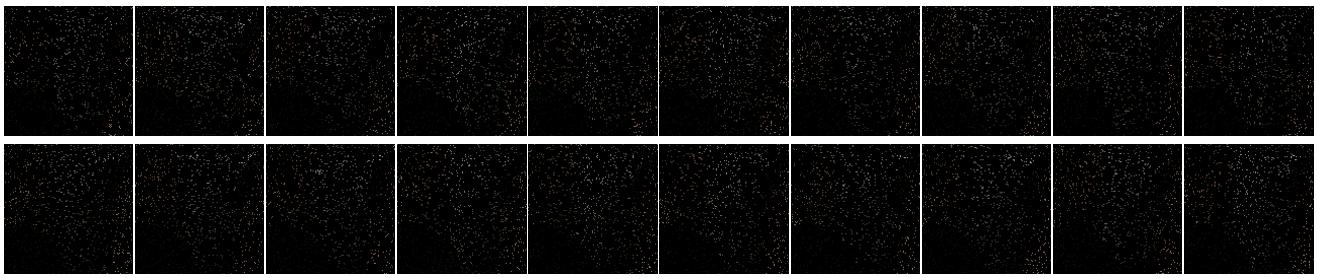


Figure 7. Masked Video generated by randomly selecting 5% of pixels without Mask Generator from ApplyEyeMakeup class.

Figure 8. Reconstructed Video generated by Frame Generator of the full model from ApplyEyeMakeup class.

Figure 9. Reconstructed Video generated by Frame Generator of the model without Mask generator with 5% of randomly selected pixels from ApplyEyeMakeup class.

Figure 10. Reconstructed Video generated by Frame Generator of the model without Mask generator with 10% of randomly selected pixelsfrom ApplyEyeMakeup class.

Figure 11. Test video to illustrate Out of Distribution imbalance of Frame Generator from Archery class.

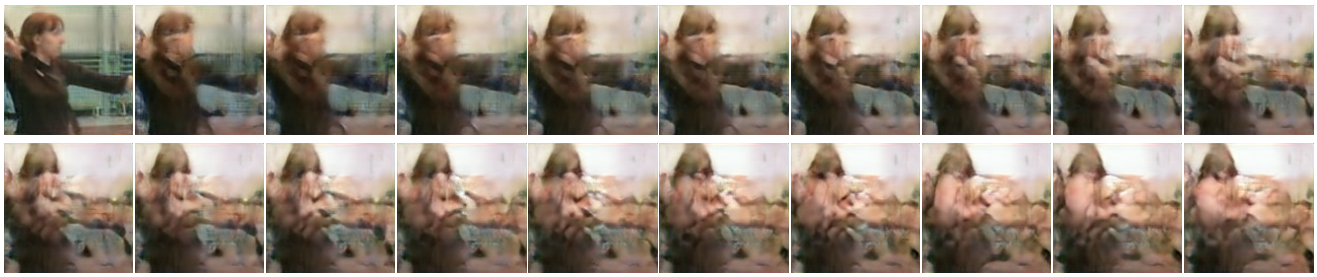


Figure 12. Reconstructed video to illustrate Out of Distribution imbalance of Frame Generator of the full model(Frames 10-20) from Archery class.

Figure 13. Reconstructed Video generated by Frame Generator of the model without Mask generator with 5% of randomly selected pixels. Reconstruction is better than the full model from Archery class.

Figure 14. Reconstructed Video generated by Frame Generator of the model without Mask generator with 10% of randomly selected pixels. Reconstruction is better than the full model from Archery class.