Towards sharper video predictions beyond traditional loss functions

Akilesh
IIT Hyderabad
cs13b1042@iith.ac.in

1. Introduction and related work

Learning to predict future images from a video sequence involves the construction of an internal representation that models the image evolution accurately, and therefore, to some degree, its content and dynamics. While optical flow has been a very well studied problem in computer vision for a long time, future frame prediction is rarely approached. Still, many vision applications could benefit from the knowledge of the next frames of videos, that does not require the complexity of tracking every pixel trajectory. In this paper, we address the problem of frame *prediction*. A major difference with the more classical problem of image reconstruction ([14], [6]) is that the ability of a model to predict future frames requires to build accurate, non trivial internal representations, even in the absence of other constraints (such as sparsity). However, lack of sharpness (or losing high-frequency components) can be observed in future frame predictions. The cause of this problem is multifold in nature, like: bad choice of loss function, architecture constraints etc. Ranzato et al. [10] defined a recurrent network architecture inspired from language modeling, predicting the frames in a discrete space of patch clusters. Srivastava et al. [13] adapted a LSTM model ([4]) to future frame prediction. However the ℓ_2 loss function inherently produces blurry results as it comes from the assumption that the data is drawn from a Gaussian distribution and works poorly with multimodal distributions. In [8], the authors address the problem of lack of sharpness in the predictions and show that generative adversarial training ([3], [1]) may be successfully employed for next frame prediction. The authors deal with blurry predictions obtained from the standard Mean Squared Error (MSE) loss function, by proposing three three different and complementary feature learning strategies: a multi-scale architecture, an adversarial training method, and an image gradient difference loss function. However, we still see further scope for improvements through appropriate modifications of loss functions. In our paper, we first explore the usage of *L_0-Regularized Intensity and Gradient Prior* method of deblurring images as proposed in [9] for the video frame predicted by [8]. We also validate the effect of using *Unnatural L_0 Sparse Representation* method suggested in [16]. We then include the above defined loss function with suitable weight as a fourth component in the combined loss expression of [8]. Combining these four losses produces the most visually satisfying results. We measure the quality of image generation by computing similarity and sharpness measures.

2. Methodology

Let $Y = \{Y^1, Y^2, \ldots Y^n\}$ be a sequence of frames to predict from input frames $X = \{X^1, X^2, \ldots X^m\}$ in a video sequence. A basic next frame prediction convnet can be trained to predict one or several concatenated frames Y from the concatenated frames X by minimizing a distance, for instance ℓ_p with p = 1 or p = 2, between the predicted frame and the true frame:

$$\mathcal{L}_p(X, Y) = \ell_p(G(X), Y) = \|G(X) - Y\|_p^p \qquad (1)$$

However, such a network has at least two major flaws:

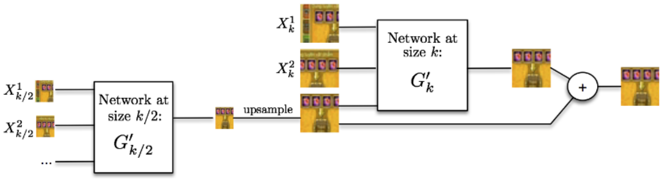
1. Convolutions only account for short-range dependencies, limited by the size of their kernels. There are a number of ways to avoid the loss of resolution brought about by pooling/subsampling while preserving long-range dependencies. The approach used in this paper is to combine multiple scales linearly as in the reconstruction process of a Laplacian pyramid [1].

2. Using an ℓ_2 loss, and to a lesser extent ℓ_1, produces blurry predictions, increasingly worse when predicting further in the future. If the probability distribution for an output pixel has two equally likely modes v_1 and v_2, the value $v_{avg} = \frac{(v_1+v_2)}{2}$ minimizes the ℓ_2 loss over the data, even if v_{avg} has very low probability. In the case of an ℓ_1 norm, this effect diminishes, but do not disappear, as the output value would be the median of the set of equally likely values.

2.1. Adversarial training

We adapted the generative adversarial network introduced by [3] and used by [8] for the purpose of frame prediction. In [3], the authors propose to use a discriminative

Figure 1: Multi-scale architecture


$X^1_{k/2}$
$X^2_{k/2}$
...
Network at size $k/2$: $G'_{k/2}$
upsample
X^1_k
X^2_k
Network at size k: G'_k
+

network D to estimate the probability that a sample comes from the dataset instead of being produced by a generative model G. The two models are simultaneously trained so that G learns to generate frames that are hard to classify by D, while D learns to discriminate the frames generated by G. Ideally, when G is trained, it should not be possible for D to perform better than chance (that is, output a score of 0.5 for every sample).

The generative model G in our case is typically the one described in the previous paragraph. The discriminative model D takes a sequence of frames, and is trained to predict the probability that the last frames of the sequence are generated by G. Note only the last frames are either real or generated by G, the rest of the sequence is always from the dataset. This allows the discriminative model to make use of temporal information, so that G learns to produce sequences that are temporally coherent with its input.

The discriminative model D is a multi-scale convolutional network with a single scalar output. The training of the pair (G, D) consists of two alternated steps, described below. For the sake of clarity, we assume that we use pure SGD (minibatches of size 1), but there is no difficulty to generalize the algorithm to minibatches of size M by summing the losses over the samples. The multi-scale architecture used in our paper is depicted in 1

Training D: Let (X, Y) be a sample from the dataset. Note that X (respectively Y) is a sequence of m (respectively n) frames. We train D to classify the input (X, Y) into class 1 and the input $(X, G(X))$ into class 0. More precisely, for each scale k, we perform one SGD iteration of D_k while keeping the weights of G fixed. It is trained with in the target 1 for the datapoint (X_k, Y_k), and the target 0 for $(X_k, G_k(X_k))$. Therefore, the loss function we use to train D is

$$\mathcal{L}^D_{adv} = \sum_{k=1}^{N_{scales}} \mathcal{L}_{bce}(D_k(X_k, Y_k), 1) + \mathcal{L}_{bce}(D_k(X_k, G_k(X_k)), 0) \tag{2}$$

where $\mathcal{L}_{bce}$ is the binary cross entropy loss, defined as

$$\mathcal{L}_{bce}(Y, \hat{Y}) = -\sum_i \hat{Y}_i log(Y_i) + (1 - \hat{Y}_i) log(1 - Y_i) \tag{3}$$

where Y_i takes its values in $\{0,1\}$ and $\hat{Y}_i$ in $[0,1]$

```
Algorithm 1: Training adversarial networks for next frame generation
Set the learning rates $\rho_D$ and $\rho_G$, and weights $\lambda_{adv}, \lambda_{\ell_p}$.
while not converged do
    Update the discriminator $D$:
    Get $M$ data samples $(X, Y) = (X^{(1)}, Y^{(1)}), \dots, (X^{(M)}, Y^{(M)})$
    $W_D = W_D - \rho_D \sum_{i=1}^{M} \frac{\partial \mathcal{L}^D_{adv}(X^{(i)}, Y^{(i)})}{\partial W_D}$
    Update the generator $G$:
    Get $M$ new data samples $(X, Y) = (X^{(1)}, Y^{(1)}), \dots, (X^{(M)}, Y^{(M)})$
    $W_G = W_G - \rho_G \sum_{i=1}^{M} \left( \lambda_{adv} \frac{\partial \mathcal{L}^G_{adv}(X^{(i)}, Y^{(i)})}{\partial W_G} + \lambda_{\ell_p} \frac{\partial \mathcal{L}_{\ell_p}(X^{(i)}, Y^{(i)})}{\partial W_G} \right)$
```

Training G: Let (X, Y) be a *different* data sample. While keeping the weights of D fixed, we perform one SGD step on G to minimize the adversarial loss:

$$\mathcal{L}^G_{adv} = \sum_{k=1}^{N_{scales}} \mathcal{L}_{bce}(D_k(X_k, G_k(X_k)), 1) \tag{4}$$

2.2. Image Gradient Difference loss (GDL)

The authors in [8] define a new loss function, the Gradient Difference Loss (GDL), that can be combined with a ℓ_p and/or adversarial loss function. The GDL function between the ground truth image Y, and the prediction $G(X) = \hat{Y}$ is given by:

$$\mathcal{L}_{gdl}(X, Y) = \mathcal{L}_{gdl}(\hat{Y}, Y) = \sum_{i,j} \left| |Y_{i,j} - Y_{i-1,j}| - |\hat{Y}_{i,j} - \hat{Y}_{i-1,j}| \right|^{\alpha} + \left| |Y_{i,j-1} - Y_{i,j}| - |\hat{Y}_{i,j-1} - \hat{Y}_{i,j}| \right|^{\alpha}$$

2.3. Combining losses

In our experiments, we combine the losses previously defined with different weights. The final loss is:

$$\mathcal{L}(X, Y) = \lambda_{adv} \mathcal{L}^G_{adv}(X, Y) + \lambda_{\ell_p} \mathcal{L}_p(X, Y) + \lambda_{gdl} \mathcal{L}_{gdl}(X, Y) \tag{5}$$

3. Experiments

We now provide a quantitative evaluation of the quality of our video predictions on UCF101 [12] and Sports1m [5]. We use 4 input frames to predict one future frame. In order to generate further in the future, we apply the model recursively by using the newly generated frame as an input. Most of the UCF101 frames only have a small portion of the image actually moving, while the rest is just a fixed background. We train our network by randomly selecting temporal sequences of patches of 32 x 32 pixels after making sure they show enough movement (quantified by the ℓ_2

difference between the frames). This is done primarily to keep the memory usage low. Since the discriminator has fully-connected layers after the convolutions, the output of the last convolution must be flattened to connect to the first fully-connected layer. The size of this output is dependent on the input image size, and blows up really quickly (e.g. For an input size of 64 x 64, going from 128 feature maps to a fully connected layer with 512 nodes, you need a connection with $64 * 64 * 128 * 512 = 268,435,456$ weights). Because of this, training on patches larger than 32 x 32 causes an out-of-memory error. The data patches are also first normalized so that their values are comprised between -1 and 1.

3.1. Evaluation Metrics

To evaluate the quality of the image predictions resulting from the different tested systems, we compute the Peak Signal to Noise Ratio (**PSNR**) between the true frame Y and the prediction $\hat{Y}$:

$$PSNR(Y, \hat{Y}) = 10 \log_{10} \frac{max_{\hat{Y}}^2}{\frac{\sum_{i=0}^{N}(Y_i - \hat{Y}_i)^2}{N}} \tag{6}$$

where $max_{\hat{Y}}$ is the maximum possible value of the image intensities.
We also measure the Structured Similarity Index Measure (SSIM) [15]. It ranges between -1 and 1, a larger score meaning a greater similarity between the two images.
To measure the loss of sharpness between the true frame and the prediction, we define the following sharpness measure based on the difference of gradients between two images Y and $\hat{Y}$:

$$Sharp.diff.(Y, \hat{Y}) = 10 \log_{10} \frac{max_{\hat{Y}}^2}{\frac{\sum_i \sum_j |(\nabla_i Y + \nabla_j Y) - (\nabla_i \hat{Y} + \nabla_j \hat{Y})|}{N}} \tag{7}$$

where $\nabla_i Y = |Y_{i,j} - Y_{i-1,j}|$ and $\nabla_j Y = |Y_{i,j} - Y_{i,j-1}|$

3.2. Results without deblurring

Table 1: Comparison of the accuracy of the predictions on 10% of the UCF101 test images

Method	PSNR	Sharpness
Our best	23.7	15.8
State-of-the-art [8]	31.5	25.4

The quantitative measures on 378 test videos from UCF101 are given in Table 2. Please note that it is the same test set as used by authors in [8]. Code is implemented in Tensorflow. It is trivial to predict pixel values in static areas, especially on the UCF101 dataset where most of the images are still, we performed our evaluation in the moving areas. Some example predictions obtained are shown in 5, 6, 7

Figure 2: Example 1 after deblurring (approach 1)

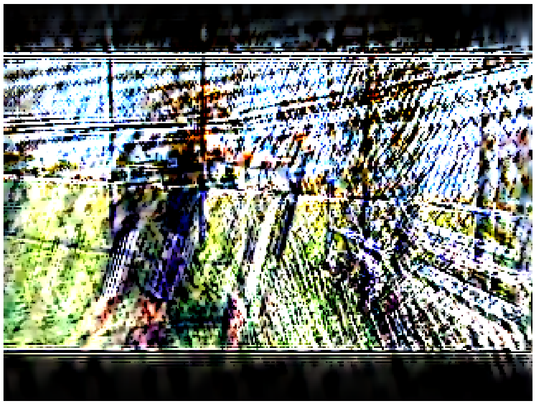
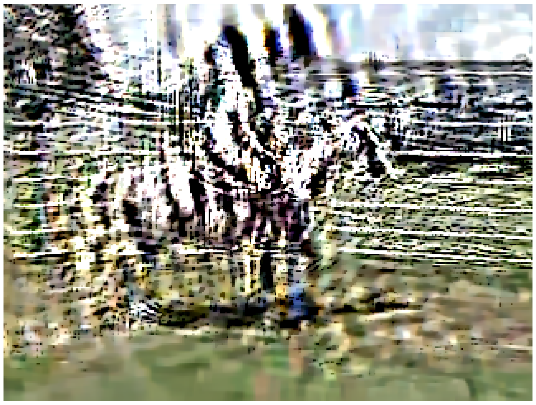


Figure 3: Example 2 after deblurring (approach 1)



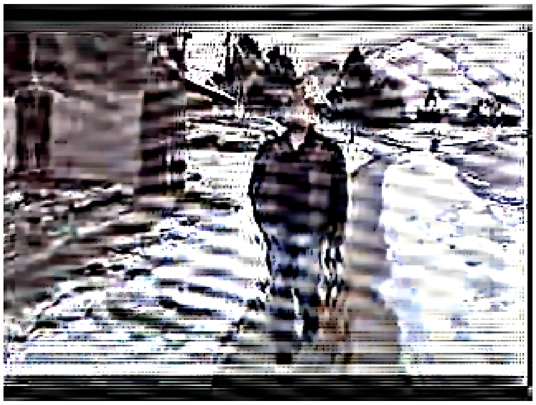
3.3. First approach for deblurring

The authors in [9] propose a simple yet effective ℓ_0-regularized prior based on intensity and gradient for text image deblurring. The proposed image prior is motivated by observing distinct properties of text images. We first explore the usage of this deblurring algorithm by passing the frames predicted by the above approach through this deblur module. As this algorithm is overly specialized for text images, it doesn't directly give good results (in fact the results are quite bad!). Some sample images obtained are shown in 8, 9, 10.

3.4. Second approach for deblurring

Next, we tried with more realistic natural image deblurring approach proposed by Xu et al. [16]. This paper proposes a generalized and mathematically sound ℓ_0 sparse expression, together with a new effective method, for motion deblurring. Significant improvements can be observed using this approach as shown in 5, 6, 7.

Figure 4: Example 3 after deblurring (approach 3)



3.5. Drawbacks with above deblurring approaches

Although the results given by natural image deblurring approach 2 is good, its a two stage approach (predict frame and then deblur). So its not end-to-end trainable. Perceptual loss propose the use of features extracted from a pretrained VGG network instead of low-level pixel-wise error measures. Specifically use a loss function based on the euclidean distance between feature maps extracted from the VGG19 network. (eg. in Super Resolution GAN paper [7]). But, since this approach has been already implemented, we decided to venture into this at a later stage.

3.6. Multiple Discriminators

Inspired by GMAN paper [2], we used two discriminators D1, D2. A multi-discriminator GAN frame-work, GMAN, allows training with the original, untampered minimax objective. The generator trains using feedback aggregated over multiple discriminators (in our case it's the mean). D1 detects whether it's real or fake and D2 detects the blurriness (the input to D2 is ground truth frame and blurred ground truth frame). We did a Gaussian blurring with 5x5 kernel and variance 5. D1, D2 are trained simultaneously and their losses are averaged before passing it to the G. Some results are shown in as shown in 8, 9, 10

Table 2: Comparison of the accuracy of the predictions on 10% of the UCF101 test images

Method	PSNR	Sharpness
Our best	23.0528	15.3786
State-of-the-art [8]	31.5	25.4

4. Future work

In the multiple discriminator approach described above, D2 doesn't have the generated image passed as input. Pretrain D2 to specialize on blur detection and then perform adversarial training. We are using 10000 images of CIFAR 10 blurred using one of the three blurring techniques: a) Gaussian blur b) Median blur c) Bilateral filtering, for training D2. We initialize weights of D2 using this pre-trained network and then follow the approach described in the previous section.

We also plan to retrain the GAN by implementing the below hacks ([11])

- Use leaky ReLU instead of ReLU to avoid sparse gradients.
- Use average pooling instead of max pooling.
- Use SGD for discriminator and Adam for generator.
- Use Batchnorm in both the generator and discriminator.
- Label Smoothing, i.e. if you have two target labels: Real=1 and Fake=0, then for each incoming sample, if it is real, then replace the label with a random number between 0.7 and 1.2, and if it is a fake sample, replace it with 0.0 and 0.3 (for example).

References

[1] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015.

[2] I. Durugkar, I. Gemp, and S. Mahadevan. Generative multi-adversarial networks. *arXiv preprint arXiv:1611.01673*, 2016.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

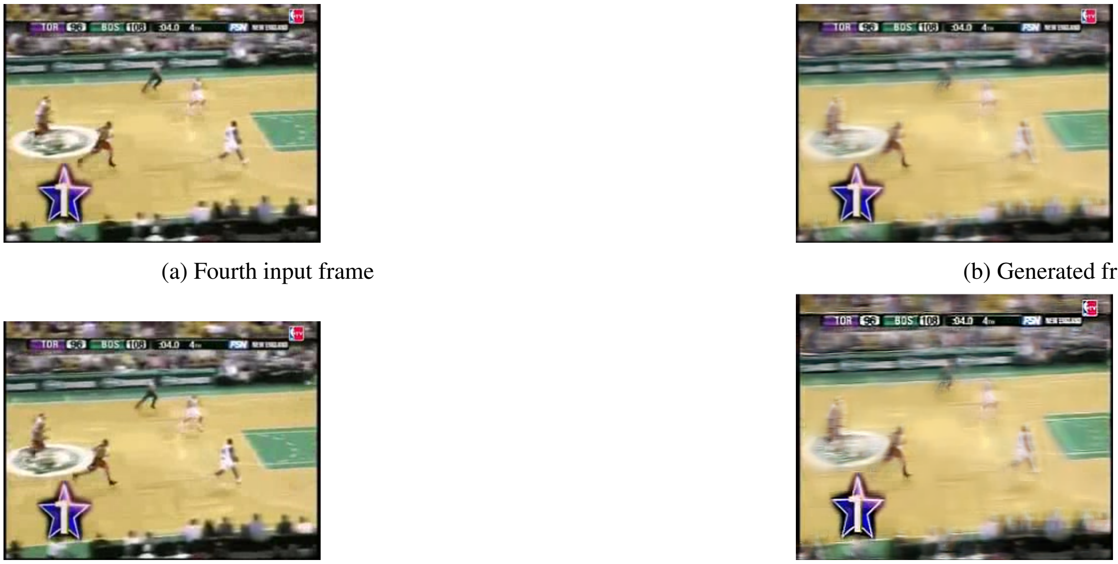
[4] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[6] Q. V. Le. Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8595–8598. IEEE, 2013.

[7] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[8] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.



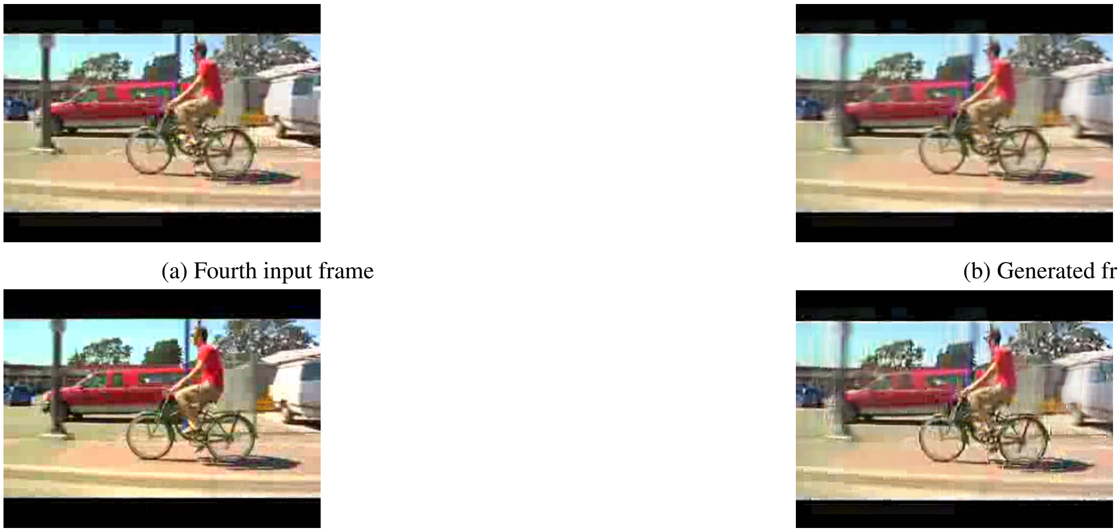
(a) Fourth input frame

(b) Generated frame

(c) Ground truth frame

(d) Deblurred (approach 2)

Figure 5: Example 1



(a) Fourth input frame

(b) Generated frame

(c) Ground truth frame

(d) Deblurred (approach 2)

Figure 6: Example 2

[9] J. Pan, Z. Hu, Z. Su, and M.-H. Yang. Deblurring text images via l0-regularized intensity and gradient prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2908, 2014.

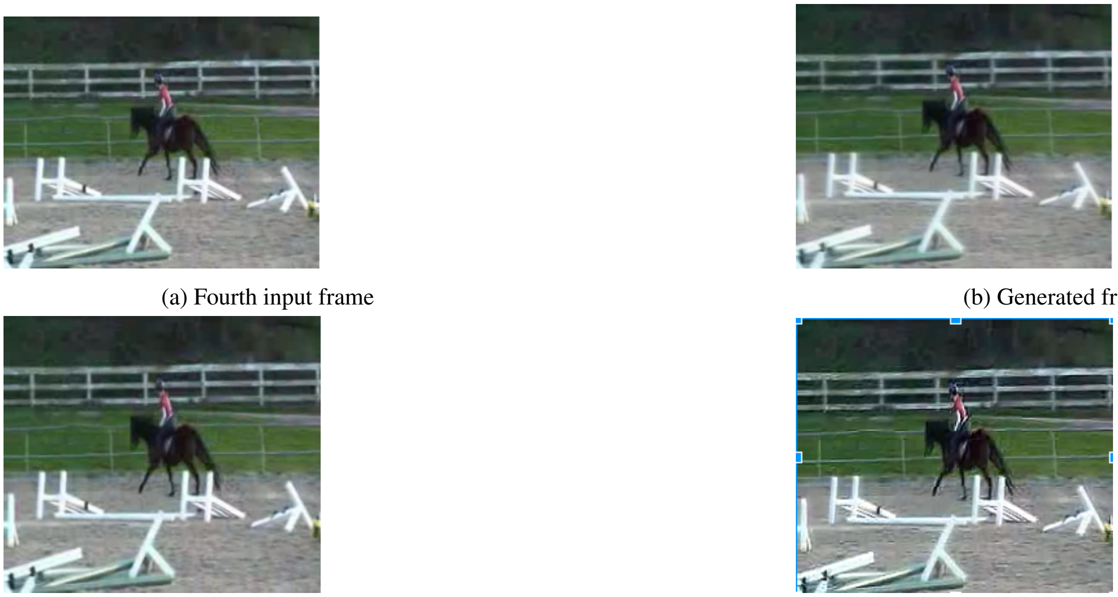
[10] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[11] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2226–2234, 2016.

[12] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[13] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*, pages 843–852, 2015.

[14] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.



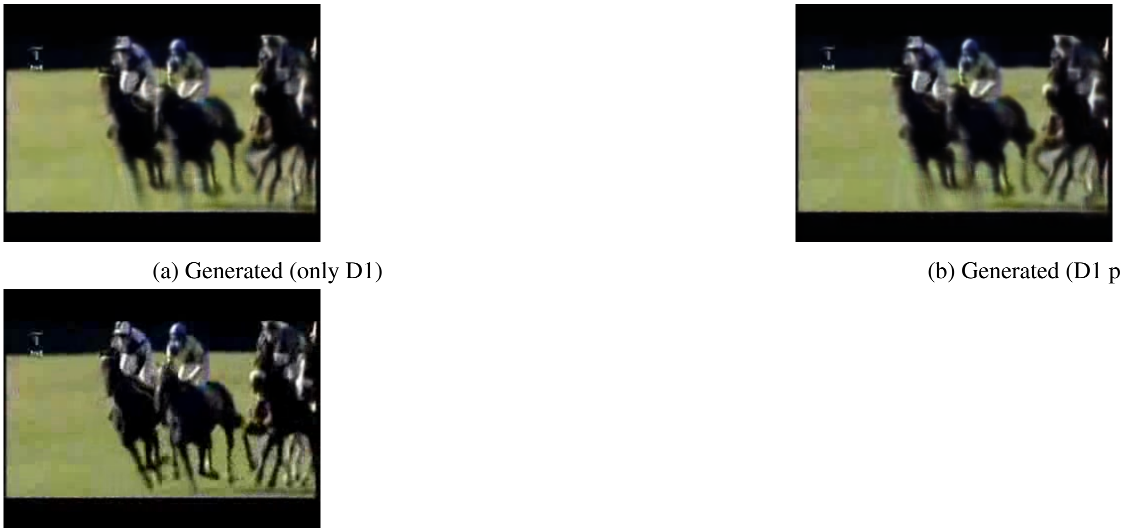
(a) Fourth input frame

(b) Generated frame

(c) Ground truth frame

(d) Deblurred (approach 2)

Figure 7: Example 3



(a) Generated (only D1)

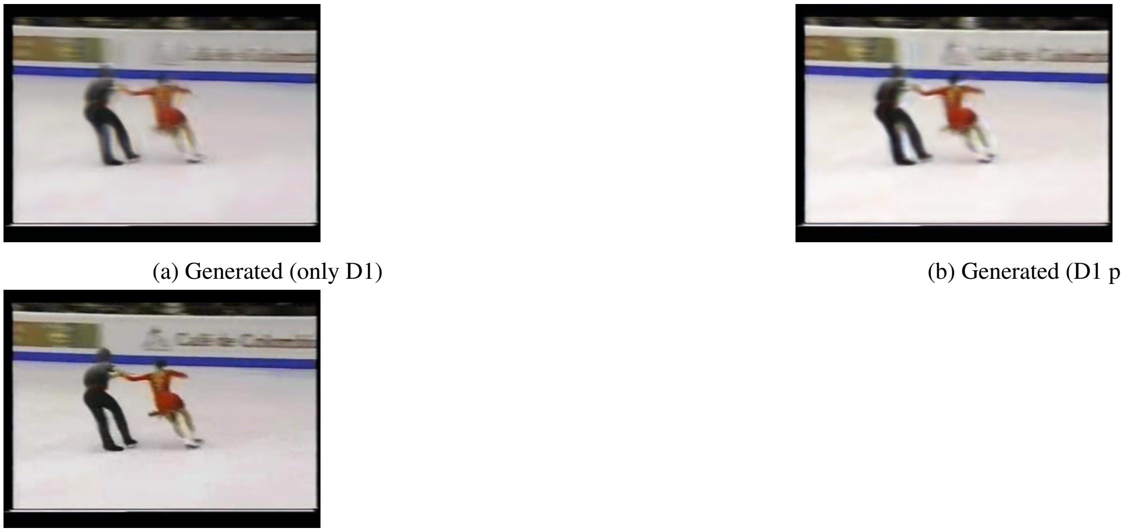
(b) Generated (D1 plus D2)

(c) Ground truth frame

Figure 8: Example 1 multiple discriminator

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

[16] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1107–1114, 2013.

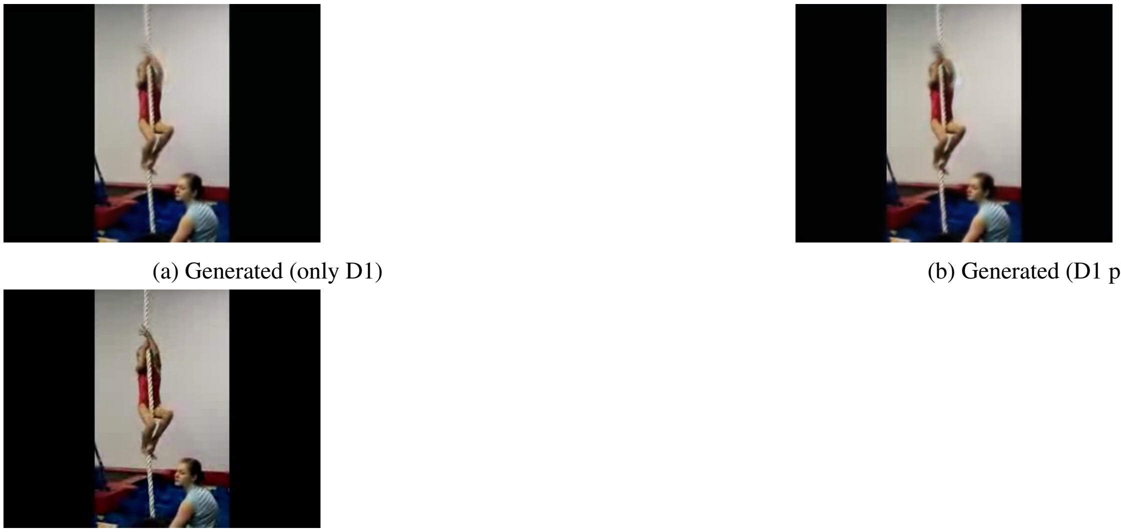


(a) Generated (only D1)

(b) Generated (D1 plus D2)

(c) Ground truth frame

Figure 9: Example 2 multiple discriminator



(a) Generated (only D1)

(b) Generated (D1 plus D2)

(c) Ground truth frame

Figure 10: Example 3 multiple discriminator