Future Action Prediction using Deep Multi-Scale Video Prediction

Shreyas Kulkarni
University of Illinois at Chicago
1200 W Harrison St
Chicago, Illinois 60607
skulka26@uic.edu

Hengbin Li
University of Illinois at Chicago
1200 W Harrison St
Chicago, Illinois 60607
hli217@uic.edu

Kruti Sharma
University of Illinois at Chicago
1200 W Harrison St
Chicago, Illinois 60607
ksharm22@uic.edu

ABSTRACT

The latest image generation model can generate images from text, new bedroom interior designs, faces etc. There are many applications where these models are used for generating novel content. Unsupervised feature learning of frames to predict next action from a video is one of the promising fields in research which requires an understanding of not just the current features of the image model but this modeling involves to predict or anticipate future features of frames accurately. This can be achieved by constructing an internal representation of video or set of images which should be able to accurately model the near future actions. This work proposes the next action prediction for a set of images of human actions (walk, slide, jump etc.) using Generative Adversarial Networks.

KEYWORDS

Generative Adversarial Networks, Video prediction, image modeling, image gradient, unsupervised feature learning

1 INTRODUCTION

What will be the next action performed by that man in the Figure: 1 ? Will that man move his hands up or down ? Predicting the next action from an image requires precision to analyze the future actions but given a set of images with some common actions, it is not very difficult to understand how the next frames in that image can be. If we have a set of images with some similar actions, tracking back these frames and modeling them together can help us to predict the next actions. The Adversarial Generator Network are widely used in generating novel designs, images and using this image reconstruction capability, we can have a network that can be used to predict the next actions from a set of frames. Combining the ability of the networks to detect objects, boundaries, scenes to predict the next action has many applications. For example, interaction with robots requires an understanding of the next actions or to be more precise understanding what part of the body will perform an action.

Unsupervised feature learning helps in learning features from unlabeled data and helps in training the network fast. Thus instead of exploiting the traditional temporal knowledge to train a 3D convolutional network or spatio-temporal convolutional network which requires long duration of training, we can use unsupervised learning and train the network using generative adversarial network. The main problem in using GAN for predicting next action is the lack of sharpness in predictions. To enhance the sharpness of action prediction, we modified the loss function making it based on image gradients which increases the sharpness and thus making the action predicted ex: a small hand movement clear in the generated image.

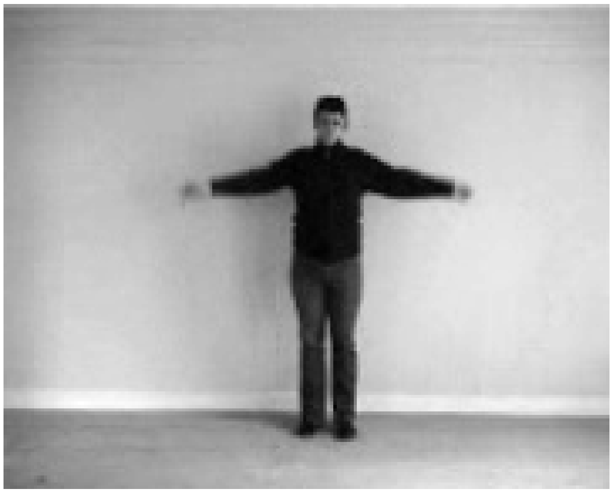

Figure 1: A sample image from Human Actions Data set of man waving

The model is referenced from paper: [7] which is based on multi-scale predictions and we try to predict the next actions from different scaled images starting from 4 X 4, 8 X 8, 16 X 16 and finally generate 32 X 32 images. The network can be defined as follows:

- Let's say we have i_1, ..., i_{N_scales} of input images where i_1 = 4 X 4, i_2 = 8 X 8, i_3 = 16 X 16 and i_4 = 32 X 32 and an upscaling factor u_s towards size i_s.
- The downscaled versions of images X_s^j and Y_s^j can be represented as $X_s^j Y_s^j$ of size i_s.
- The network G_s learns to predict Y_s - $u_s(Y_{s-1})$ from X_s and guesses a coarse of Y_s. Thus we can define our network recursively to make a prediction Y_s of size i_s as:

$$Y_s = G_s(X) = u_s(Y_{s-1}) + G_s\Big(X_s, u_s(Y_{s-1})\Big)$$

- Thus the network is modeled to make a series of prediction starting from the lowest scale or resolution images.

2 RELATED WORK

There are various papers and algorithms that have explored the next action/frame prediction using different networks and training. Below are some of the related work which we have referenced:

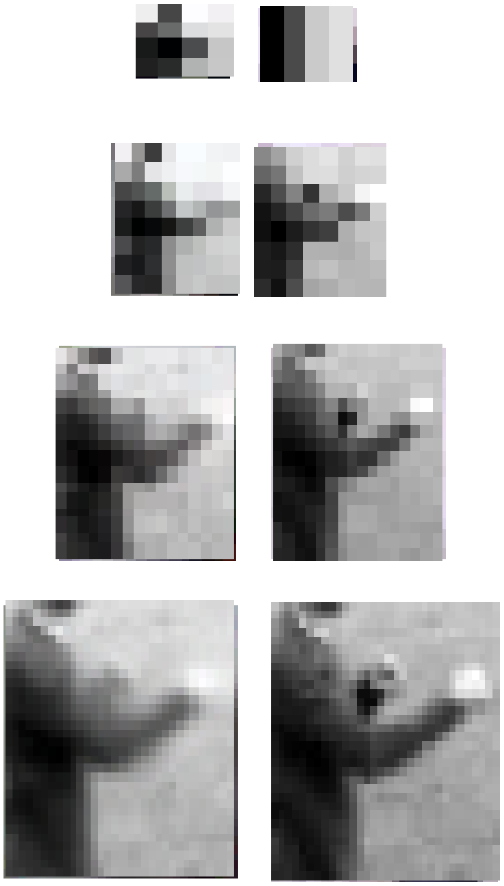

Figure 2: Generated vs Ground Truth for different scale sizes.

- **3D Convolutional Neural Networks for Human Action Recognition** [5] exploits the spatial and temporal information in a supervised way for action recognition. The paper uses Convolutional Neural Networks which can act directly on the raw inputs which helps in automating the process of feature construction. By using the spatial and temporal information to extract features, this model can capture the motion information encoded in multiple adjacent frames.
- **Invariant recognition drives neural representations of action sequences** [8] is another spatiotemporal Convolutional Neural Network which categorizes video stimuli into actions i.e. recognizing actions of others from visual stimuli. As per their model, the invariant action recognition task can be used to represent data which can match human neural recordings.

  The problem in using above convolutional networks is the amount of time in training the model for supervised learning. Thus a new model consisting of convolutional network with unsupervised feature learning can help in reducing the training time and predict the future frames with more sharpness and accuracy.

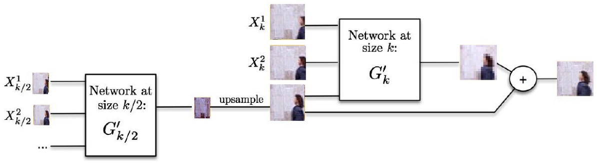
$X^1_{k/2}$
$X^2_{k/2}$
...
Network at size $k/2$: $G'_{k/2}$
upsample
X^1_k
X^2_k
Network at size k: G'_k
+

Figure 3: Architecture of Network

- **PredNet** [6] is a deep convolutional recurrent network which is trained for next-frame video prediction with the belief that prediction is an effective objective of unsupervised or self-supervised learning. The model was trained by feeding the predictions recursively to make Multi-timestep ahead predictions.
- **Anticipating the future by watching unlabeled video** [10] uses a deep convolutional network which learns to anticipate both actions and objects in the future by using a large scale framework that capitalizes on temporal structure in unlabeled video. Their network architecture produces multiple prediction which helps to understand the actual prediction.
- **Generating Videos with Scene Dynamics by Vondrick** [11] is one of the convolutional model that learns scene dynamics from a large amount of unlabeled videos for action classification and future prediction. This paper proposed a generative adversarial network for video with a spatio-temporal convolutional architecture that untangles the scene's foreground from the background. There are many other models which helps in predicting future video frames, for example: predicting frames in a discrete space of patch clusters using recurrent neural network inspired from language modeling by Ranzato et. al. The adaptation of LSTM model by Srivastava, action conditional auto-encoder by Oh et al(2015) helped in predicting next action in Atari-like games.

  The above architectures and models produces an accurate prediction but with blur effects and lacks sharpness in the predicted frames. Our work is inspired from *Generating Videos with Scene dynamics* and proposes a new architecture to improve the predicted image sharpness by modifying the loss functions of generative adversarial network which helps in preserving the sharpness of the frames. The model, architecture and loss fucntions presented in this paper are referenced from the paper **Deep Multi-Scale Video Prediction Beyond Mean Square Error** [7].

3 ARCHITECTURE

A traditional *Convolutional Network* [9] with alternating convolutions and *Rectified Linear Units (ReLU)* [1] network suffers from some flaws, for example: since the size of the kernel is limited, the convolutions can account only for short-range dependencies . If we try to avoid the short-range dependencies using pooling/subsampling, we encounter the problem of resolution loss. Thus avoiding pooling/subsampling and still keeping the long-range dependencies can be achieved by using multiple convolution layers or use the connections to skip the pooling/unpooling pairs. Thus in order to avoid the resolution loss and preserve the high-frequency information, the new model proposed is to combine multiple scales linearly same as used in the reconstruction process of a Laplacian pyramid [2]. The **Multi-Scale Network** is a multi-scale model where the network makes a series of prediction starting from the lowest resolution and uses the prediction of one size (smaller size) as a starting point to make the prediction of a size greater than the original and the overall minimization is performed via *Stochastic Gradient Descent (SGD).*

3.1 Adversarial Training

This work is based on the idea presented in paper: [7] and uses the Generative Adversarial Network introduced by Goodfellow where image patches are generated from random noise using two networks trained simultaneously. The Discriminator network D is used to predict or estimate the probability that an output image produced belongs to the original dataset or is being produced by the Generative model G. The two models : D and G are simultaneously trained so that G learns to generate the frames that hard to classify by D, whereas D learns to discriminate the frames generated by G. This approach is adapted into predicting next frames.

Generative Model G: is the same as used proposed in the original paper which learns the joint probability of the input data and labels simultaneously. Since we are using unlabeled videos, the model only learns the input data without any labels to generate new images from input data.

Discriminative model D: learns a function that maps the input data to some desired output class. Here we change the model in a way that the Generator model only generates the last image of the sequence, so the discriminator model is trained to predict if the last frame from a sequence of frames (provided as input) is real or generated by G. The models task to discriminate only the last frame allows it to make use of temporal information thus G learns to produce sequences that are temporally coherent with its input dataset. As the generator model now depends on the sequence and the temporal information of input images, the training for G can be done in the absence of random noise. This model is a multi-scale convolutional network with a single scalar output.

For understanding the training of both the models, let's say we want to predict a set of next frames say Y where Y = (Y^1,.....,Y^n) from a sequence of input frames X, where X = (X^1,.....,X^m), we can train our models G and D for a sample (X,Y) from dataset and we assume that we are using pure SGD on minibatches of size 1 which can be later generalized for any minibatch size M by summing the loss over the samples.

Training G: In order to train the generative model, we assume that the weights used in D are fixed, when we perform one minimization using SGD on G, it helps to reduce the adversarial loss thus making the discriminative model D more confused. So now as this loss is reduced and D will not be able to discriminate in the prediction correctly. The adversarial loss can be calculated as:

$$L_a^D dv(X,Y) = \sum_{k=1}^{N_scales} L_bce(D_k(X_k, G_k(X_k)), 1)$$

where L_bce is the binary cross-entropy loss which is defined as: $L_{bce}(Y,Y) = -\sum_i \hat{Y}_i log(Y_i) + (1 - \hat{Y}_i log(1 - Y_i)$

where Y_i has its value in range of 0,1 and $\hat{}Y_i$ in [0,1]

But this minimizing alone is not sufficient to make the discriminator model confused, because G can always produce confusing samples without being close to Y, making the discriminator model to learn and discriminate these samples. This will lead to generator model to generate more random samples which can cause instability. To tackle this problem, we introduce another loss: L_p and the generator is trained to minimize the combined adversarial and L_p loss.

Training D: Training of D is to classify the input (X,Y) into class 1 and (X,G(X)) into class 0 i.e. while we keep the weights of G fixed, we perform one SGD iteration of D_k for each scale k and we train (X_k,Y_k) within the target 1 and (X_k,$G_k(X_k)$) for target 0. The loss function used for training D is:

$$L_{adv}^D(X,Y) = \sum_{k=1}^{N_scales} L_bce(D_k(X_k, Y_k), 1) + L_bce(D_k(X_k, G_k(X)), 0)$$

Algorithm We have used the concept and algorithm provided in the paper: [7].

- **Pre-Processing** Before training the network, all the unlabeled video clips are converted into images and each set of images for a video are maintained in a separate directory. The pre-processing of these images involves converting a set of input images (ex:5) into a numpy array and finally compressing all the numpy array for a given set of inputs (5) into uncompressed *npz* format.
- Initialize the learning rates for ρ_D and ρ_G, weights λ_{adv}, λ_{l_p}.
- **Training** of discriminator and generator involves to train while both networks do not converge, we repeat updating first the Discriminator model and then the Generator model.
- **Update Discriminator:** For M data samples (any batch size within a range of 8 gives good results) (X,Y) we have ($X^{(1)}$,$Y^{(1)}$, ...,($X^{(M)}$,$Y^{(M)}$ and weights are updated as:

$$W_D = W_D - \rho_D \sum_{i=1}^{M} \frac{\partial L_{adv}^D(X^{(i)}, Y^{(i)})}{\partial W_D}$$

- **Update Generator:** Similarly we update the weights of generator model for M new data samples as:

$$W_G = W_G - \rho_G \sum_{i=1}^{M} \left(\lambda_{adv} \frac{\partial L_{adv}^G(X^{(i)}, Y^{(i)})}{\partial W_G} + \lambda_{l_p} \frac{\partial L_{l_p}(X^{(i)}, Y^{(i)})}{\partial W_G} \right)$$

- **Final Loss Function:** The final loss function is computed by adding adversarial loss, L_p loss and the image gradient difference loss (gdl). The **Image Gradient Difference loss (GDL)** is used to sharpen the image prediction where the differences of image gradient predictions are directly subtracted from the generative loss function.

4 EXPERIMENTS

4.1 Datasets

We used the data set from **Actions as Space-Time Shapes** [4] which has a database of 90 low-resolution video sequences and has nine different people performing different natural actions as: "run", "skip", "slide", "bend" etc. We converted each of the video into frames and the network is trained on randomly selected images from a batch with image size cropped to 32 X 32 pixels. The normalization of the data patches changes the values to range between -1 and 1 and compressing it save the numpy array of multiple images. The various actions in the data set are:

- Bend
- Boxing
- Handclapping
- Handwaving
- Jogging
- Running
- Side
- Skip
- Walk

4.2 Evaluations

In order to quantify the results, we trained the network for 100,000 iterations and as per the losses defined in the paper: [7], we observed the *Global Loss (GL)*, *Peak Signal to Noise Ratio(PSNRE)* and *Sharpness Difference Error (SDE)* across iterations ranging from 1000 to 100,000 for Discriminator and Generator for our trained data set. Table 1 shows the values of Global Loss, PSNR Error and Sharpness Difference Error across range of 1000 - 100,000 for both Discriminator and Generator.

PSNR Calculation:

$$PSNR(Y, Y') = 10\log_{10}\frac{max'^2_Y}{\frac{1}{N}\sum_{i=0}^{N}(Y_i - Y'_i)^2}$$

Sharpness Difference Error:

$$Sharp.diff(Y, Y') = 10\log_{10}\frac{max'^2_Y}{\frac{1}{N}(\sum_i\sum_j |(\nabla_i Y + \nabla_j Y) - (\nabla_i Y' + \nabla_j Y')|)}$$

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

4.3 Results

The implementation of this GAN was referenced from [3]. After training the network for 100,000+ iterations, following were the prediction made from the trained data set. We can see from the results that how the sharpness of the images decreases with the increase in number of predictions. But still the network can predict upto four frames with high accuracy.

Table 1: Observation of Global Loss(GL), Peak Signal to Noise Ratio Error(PSNRE) and Sharpness Difference Error(SDE) across different iterations

Steps	Discriminator	Generator
1000	GL: 4.816291	GL: 193.038 PSNRE: 22.0620 SDE: 13.9162
2,000	GL: 4.816044	GL: 217.500 PSNRE: 18.8502 SDE: 13.4680
4,000	GL: 4.808631	GL: 511.017 PSNRE: 18.4962 SDE: 12.4848
8,000	GL: 5.703743	GL: 466.118 PSNRE: 17.3933 SDE: 11.901
10,000	GL: 5.703678	GL: 518.445 PSNRE: 22.0620 SDE: 12.6583
50,000	GL: 5.701851	GL: 87.5287 PSNRE: 26.2858 SDE: 15.2184
1,00,000	GL: 4.971539	GL: 162.328 PSNRE: 24.7812 SDE: 15.6159

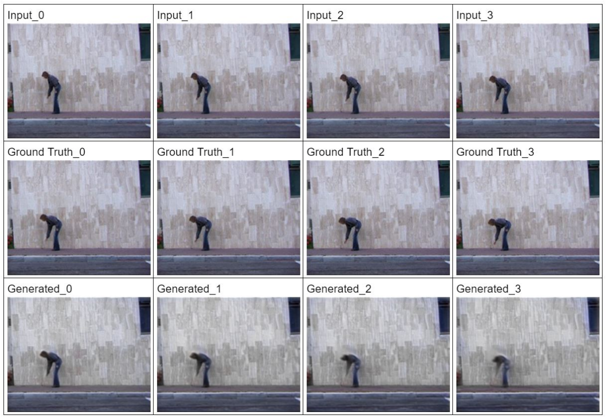

Figure 4: Results with next four predictions for action as Bend

In Figure: 4 , we have shown the first 4 input images that the network takes as input, in order to compare the generated images and their ground truths, we show 4 ground truths and 4 predictions. The last predicted frame has less sharpness but the predicted frame still has a good accuracy.

In Figure: 5 , we have shown the first 4 input images that the network takes as input and here we generate only 3 predictions. The last prediction still has a better sharpness as compared to the fourth prediction in Figure: 4

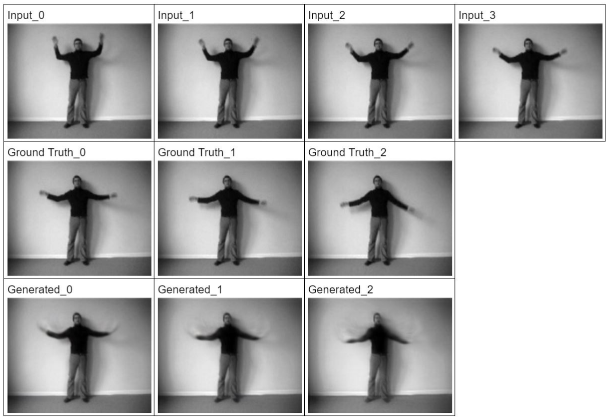

Figure 5: Results with next three predictions for action as Wave

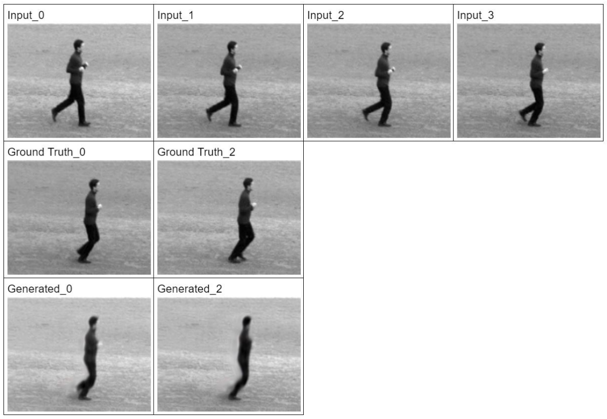

Figure 6: Results with next four predictions for action as Run

In Figure: 6 and Figure: 7 , we have shown the first 4 input images that the network takes as input and the number of predictions made are only 2 and 1 frames. The accuracy and sharpness for the only one prediction is better as compared to any other predicted frames. Overall the sharpness precision has increased using GAN but further enhancements can still be made for better accuracy.

5 CONCLUSIONS

The results for predicting next frames for different actions have high accuracy and sharpness and using Generative Adversarial Networks helps to decrease the overall training time. The multi-scale architecture has an advantage over a 3D convolution network or any other spatio-temporal recurrent network in predicting the future frames with good precision. The future work for this architecture can be to enhance the performance and classify the features in a weakly supervised context. This architecture can be used as a replacement for models where next frame predictions are required for example in it can be further extended for models where the next frame is unknown in video streams.

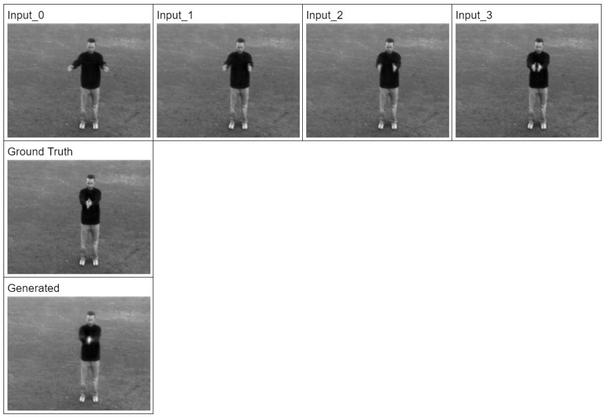

Figure 7: Results with next one predictions for action as Clap

A APPENDIX

A.1 Introduction

A.2 Related Work

A.3 Architecture

A.3.1 Adversarial Training.

Generative Model G.

Discriminator Model D.

Training G.

Training D.

Algorithm.

A.4 Experiments

A.4.1 Datasets.

A.4.2 Evaluations.

A.4.3 Results.

A.5 Conclusions

A.6 References

ACKNOWLEDGMENTS

We would like to acknowledge the work presented in the paper: [7]. We verified the results as presented in the paper on the data set of Human Actions and predicted the next action frames i.e. predicting the next movement or action performed by human. We would like to thank Professor Angus Forbes for giving us such an interesting and challenging project. The amount of resources and help that we have received from all the documentation shared by him has been of great help.

REFERENCES

[1] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. 2016. Understanding Deep Neural Networks with Rectified Linear Units. *CoRR* abs/1611.01491 (2016). http://arxiv.org/abs/1611.01491

[2] Emily L. Denton, Soumith Chintala, Arthur Szlam, and Robert Fergus. 2015. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. *CoRR* abs/1506.05751 (2015). http://arxiv.org/abs/1506.05751

[3] Github 2016. *Adversarial Video Generation*. Github. https://github.com/dyelax/Adversarial_Video_Generation.

[4] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. 2007. Actions as Space-Time Shapes. *Transactions on Pattern Analysis and Machine Intelligence* 29, 12 (December 2007), 2247–2253.

[5] S. Ji, W. Xu, M. Yang, and K. Yu. 2013. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 1 (Jan 2013), 221–231. https://doi.org/10.1109/TPAMI.2012.59

[6] William Lotter, Gabriel Kreiman, and David Cox. 2016. Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning. *CoRR* abs/1605.08104 (2016). http://arxiv.org/abs/1605.08104

[7] Michaël Mathieu, Camille Couprie, and Yann LeCun. 2015. Deep multi-scale video prediction beyond mean square error. *CoRR* abs/1511.05440 (2015). http://arxiv.org/abs/1511.05440

[8] A. Tacchetti, L. Isik, and T. Poggio. 2016. Invariant recognition drives neural representations of action sequences. *ArXiv e-prints* (June 2016). arXiv:q-bio.NC/1606.04698

[9] UFLDL 2013. *Convolutional Neural Network*. UFLDL. http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork.

[10] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2015. Anticipating the future by watching unlabeled video. *CoRR* abs/1504.08023 (2015). http://arxiv.org/abs/1504.08023

[11] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating Videos with Scene Dynamics. In *NIPS*.