Next Frame Prediction

Aditya Ramesh Ogale
NetID: 678453407
aogale3@uic.edu

Abstract

Next frame prediction is the task of predicting the future frames based on few initial given frames. This task is equally important and cumbersome to implement. In this project, I propose use of Vision Transformer based Generative Adversarial Networks, LSTM networks and Hourglass model to generate the final image conditioned on the high-level structure, i.e. the pose. Hourglass model gives the high-level structure, the pose, LSTM network predicts the future pose and the ViTGAN synthesizes the image from the predicted pose. Use of ViTGANs improves the quality of image synthesis significantly.

1 Introduction

Predicting what happens next in the future is one of the most important problems of the next generation of machine learning tasks. One such problem is predicting what happens next in a video given a set of initial images i.e. the prediction and anticipation of future events where the input of the network is the previous few frames, and prediction is/are the next frame(s). Even though this problem seems quite easy to the humans, it is extremely challenging from a machine's point of view.

Modeling contents and dynamics from videos or images is the main task for next-frame prediction which is different from motion prediction. Next-frame prediction is to predict future image(s) through a few previous images or video frames whereas motion prediction refers to inferring dynamic information such as human motion and an object's movement trajectory from a few previous images or video frames. Many examples of predictive systems can be found where next-frame prediction is beneficial. For instance, predicting future frames enables autonomous agents to make smart decisions in various tasks. Kalchbrenner et al.[3] proposed a video pixel network that contributes to helping robots make decisions by understanding the current images and estimating the discrete joint distribution of the raw pixel values between images. Other approaches provided a visual predictive system for vehicles, which predicts the future position of pedestrians in the image to guide the vehicles to slow down or a brake.

Since deep learning has shown its effectiveness in image processing, deep learning for next-frame prediction is very powerful compared with traditional machine learning. It is difficult to learn the features from images efficiently. This makes the traditional machine learning approaches cumbersome as they require the manual extraction of features and much preprocessing work. In this regard, deep learning is relatively easier. Recent approaches include pixel-level video prediction which highly depends on observing the generated frames in the past to make predictions further into the future. To make long-term predictions, these approaches need to be highly robust to pixel-level noise. However, the noise amplifies quickly through time until it overwhelms the signal making the images generated blurry and of very low quality until the context of the video is lost.

There has been a lot of research and novel approaches in the field of next-frame prediction. Michael Mathieu et al., 2016 [5] proposed a pyramid of CNNs and a GAN to predict the next frame. To deal with the inherently blurry predictions obtained from the standard Mean Squared Error (MSE) loss function, they proposed three different and complementary feature learning strategies: a multi-scale architecture, an adversarial training method, and an image gradient difference loss function.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Srivastava et al., 2016 [9] proposed LSTM networks to learn representations of video sequences and encoder LSTM to map an input sequence into a fixed-length representation. This representation was decoded using single or multiple decoder LSTMs to perform predicting the future sequence. Vukotić et al., 2017 [11] proposed an encoder-decoder model using CNNs with 2 branches. One branch takes an encoded image as the input and the other one takes as input an arbitrary time difference to the desired prediction. Oliu et al., 2018 [7] introduced bGRU and proposed a model consisting of CNN and GRU autoencoders. One of the drawbacks of these methods was that the length of the output sequence was very small (maximum of 20).

Ruben Villegas et al., 2018 [10] proposed a hierarchical approach to making long-term predictions of future frames that involved generative modeling of video using high-level structures. The proposed algorithm first estimates high-level structures of observed frames and then predicts their future states, and finally generates future frames conditioned on predicted high-level structures. This approach generates up to 128 images into the future with the input of just 10 images.

Höppe et al., 2022 [2] proposed a diffusion model RaMViD which extends image diffusion models to videos using 3D convolutions and introduces a new conditioning technique during training. The model can be summarized as a novel diffusion-based architecture for video prediction and infilling, competitive performance with recent approaches across multiple datasets, and the introduction of a schedule for random masking.

Recently, Vision Transformers (ViTs) have shown competitive performance on image recognition while requiring less vision-specific inductive biases. Lee et al., 2021 [4] investigated if such observation can be extended to image generation. The research showed that with appropriate tweaks, the proposed model ViTGAN produces far better results computationally and quality-wise. In this project, I explored the use of ViTGANs to generate images after predicting the high-level structure for the given time step using LSTMs.

2 Background

The system proposed by Ruben Villegas et al[10]. 2018 consisted of a pipeline with the following components 1) performing high-level structure estimation from the input sequence, 2) predicting a sequence of future high-level structures and 3) generating future images from the predicted structures by visual-structure analogy-making given an observed image and the predicted structures. The high-level structure, in this case, poses, are extracted using the Hourglass model proposed by Newell et al., 2016 [6]. The Hourglass model is used for pose estimation on input images extracted by stacking multiple hourglass networks of the encoder-decoder model with skip-connections. Subsequently, a sequence-to-sequence LSTM-recurrent network is trained to read the outputs of the Hourglass network and to predict the future pose sequence. Finally, future frames are generated by analogy-making using the pose relationship in feature space to transform the last observed frame. The visual-structure analogy was inspired by Reed et al. (2015) [8] following A:B::C:D, read as "A is to B as C is to D". The model tries to apply the same transformation to image C that is applied to image A to generate image B. Applying this set of transformations generate the image D. Figure 1 illustrates this hierarchical approach.

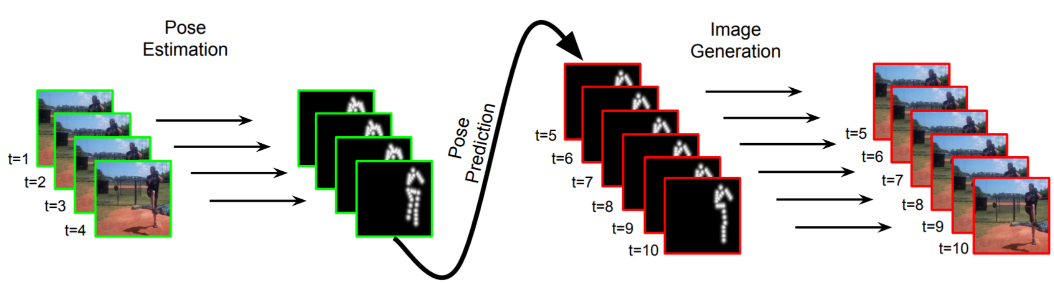


Figure 1: Hierarchical approach to pixel-level video prediction

In this approach, we have 3 components, **Pose Estimator**: Estimates the pose from the image using the Hourglass model, **Pose Predictor**: The poses estimated are passed through an LSTM network to predict future poses, **Frame Generator**: The pose predicted is passed through an analogy-making encoder-decoder model.

One of the major drawbacks in previous attempts was that the images generated were of low quality after passing through an RNN. This drawback occurred as the subsequent predictions used the poses generated in the previous timestep. The prediction error that emerged during the prediction of a pose, was propagated through the network to the subsequent poses and amplified the impact of these losses. In this architecture, for subsequent predictions, the LSTM does not observe previously generated poses and predictions are generated only from the original observations.

As in the analogy-making model, image generation has 3 parts, **Pose encoder**: Convolutional encoder which gives the high-level human structure, **Image encoder**: Convolution encoder which converts observed appearance into the features, **Image decoder**: convolutional encoder which synthesizes future frames from the poses.

In this approach, we train our high-level structure LSTM independent of the visual-structure analogy network, but both are combined during test time to perform video prediction. To train our network, the compound loss was used inspired by Dosovitskiy Brox (2016) [1] and given by,

$$\mathcal{L} = \mathcal{L}_{img} + \mathcal{L}_{feat} + \mathcal{L}_{Gen}$$

In the paper published by Lee et al. 2021 [4], the use of a Vision Transformer to train the GANs without any convolutional layer or pooling layers is explored. One of the challenges is that the training becomes unstable when the GAN is coupled with the Vision Transformer with high variance gradients. Also, conventional penalties like Gradient Penalty and spectral normalization do not work on this issue. This problem is then solved by using the Lipschitz property of self-attention and a novel architecture of the generator.

Lipschitz continuity is violated in Vision Transformers when the Lipschitz constant of the dot product is used in the self-attention layer. To counter this, L2 attention is adopted where the dot product is replaced by the L2 distance d. Applying increased spectral normalization also helps stabilize the training. To avoid overfitting, and Vision Transformer not focusing on local cues and providing meaningful loss to the generator, an overlap of zero-pixels is used for each patch.

Three new architectures for the generator were also proposed. In the first architecture, the generator takes the input of a sequence of positional embeddings and adds a latent vector w to each of the positional embeddings. In the second architecture, instead of adding the latent vector w to each positional embedding, prepends it to the entire sequence. In the third approach, instead of sending w to the transformer, it is first fed to the self-modulated LayerNorm. In the output mapping, the output of the transformer is first coupled with the Fourier features, in this case, the sine activation function. This makes the pixel values lie between -1 to 1. Each patch is generated by two MLP layers for the synthesized patch. This architecture can be seen in Fig. 2. Here, the A depicts first architecture, B depicts second and C depicts the third architecture which will be used.

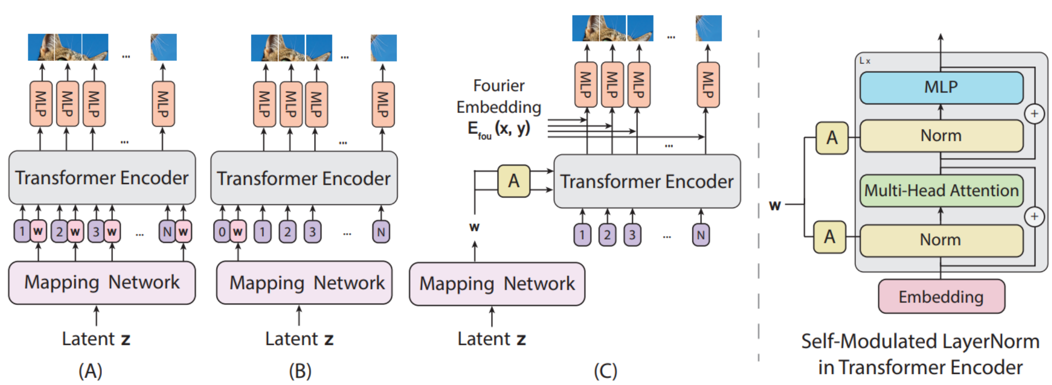
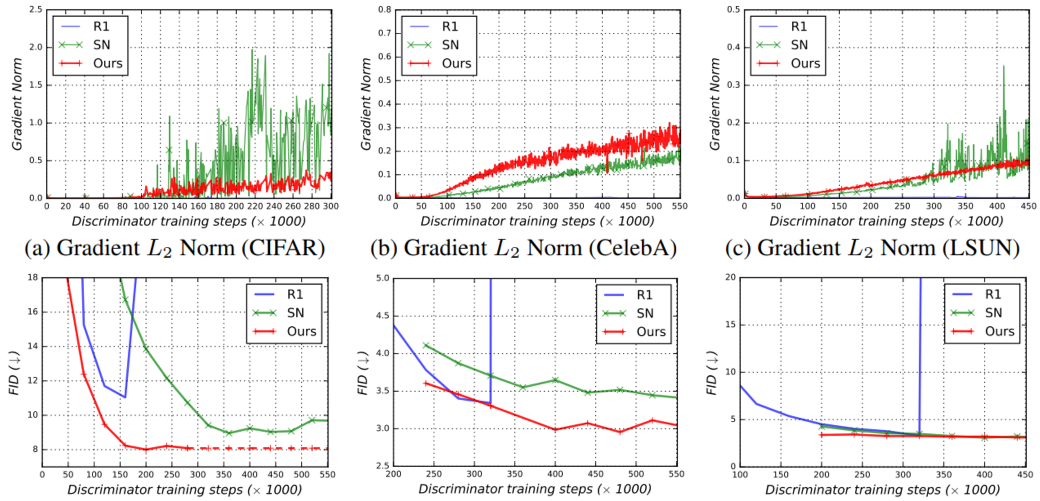


Figure 2: Hierarchical approach to pixel-level video prediction

ViTGAN model outperforms other Transformer-based GAN models by a large margin. This results from the improved stable GAN training on the Transformer architecture, as shown in Fig. 3. It achieves comparable performance to the state-of-the-art CNN-based models.



(a) Gradient L_2 Norm (CIFAR) (b) Gradient L_2 Norm (CelebA) (c) Gradient L_2 Norm (LSUN)

(d) FID (CIFAR) (e) FID (CelebA) (f) FID (LSUN)

Figure 3: Hierarchical approach to pixel-level video prediction

3 Method

In this project, I propose a hierarchical model in which the image is first converted to a high-level structure, a pose. The Hourglass model can be used for this task. Using this pose and time steps t, build an LSTM network that will output the future poses. Note that, in this approach as well, the LSTM network does not observe the poses generated in the previous time steps. The image generation can be done using ViTGAN. Input to the ViTGAN will be the noise conditioned on the pose generated, and the output will be the image generated which the discrimination will try to distinguish if the image is fake or not.

In future work, instead of estimating the pose of all images, then predicting the future poses and synthesizing the image to finally generate the video, one can use VAEs to convert the image to a latent space z and using the same LSTM network, can predict the latent variable z. This latent variable z will be the input to the ViTGAN.

References

[1] Alexey Dosovitskiy and Thomas Brox. "Generating Images with Perceptual Similarity Metrics based on Deep Networks". In: *CoRR* abs/1602.02644 (2016). arXiv: `1602.02644`. URL: `http://arxiv.org/abs/1602.02644`.

[2] Tobias Höppe et al. *Diffusion Models for Video Prediction and Infilling*. 2022. DOI: `10.48550/ARXIV.2206.07696`. URL: `https://arxiv.org/abs/2206.07696`.

[3] Nal Kalchbrenner et al. "Video Pixel Networks". In: *CoRR* abs/1610.00527 (2016). arXiv: `1610.00527`. URL: `http://arxiv.org/abs/1610.00527`.

[4] Kwonjoon Lee et al. "ViTGAN: Training GANs with Vision Transformers". In: *CoRR* abs/2107.04589 (2021). arXiv: `2107.04589`. URL: `https://arxiv.org/abs/2107.04589`.

[5] Michael Mathieu, Camille Couprie, and Yann LeCun. *Deep multi-scale video prediction beyond mean square error*. 2015. DOI: `10.48550/ARXIV.1511.05440`. URL: `https://arxiv.org/abs/1511.05440`.

[6] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked Hourglass Networks for Human Pose Estimation". In: *CoRR* abs/1603.06937 (2016). arXiv: `1603.06937`. URL: `http://arxiv.org/abs/1603.06937`.

[7] Marc Oliu, Javier Selva, and Sergio Escalera. "Folded Recurrent Neural Networks for Future Video Prediction". In: *CoRR* abs/1712.00311 (2017). arXiv: 1712.00311. URL: http://arxiv.org/abs/1712.00311.

[8] Scott E Reed et al. "Deep Visual Analogy-Making". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper/2015/file/e07413354875be01a996dc560274708e-Paper.pdf.

[9] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. "Unsupervised Learning of Video Representations using LSTMs". In: *CoRR* abs/1502.04681 (2015). arXiv: 1502.04681. URL: http://arxiv.org/abs/1502.04681.

[10] Ruben Villegas et al. "Learning to Generate Long-term Future via Hierarchical Prediction". In: *CoRR* abs/1704.05831 (2017). arXiv: 1704.05831. URL: http://arxiv.org/abs/1704.05831.

[11] Vedran Vukotic et al. "One-Step Time-Dependent Future Video Frame Prediction with a Convolutional Encoder-Decoder Neural Network". In: *CoRR* abs/1702.04125 (2017). arXiv: 1702.04125. URL: http://arxiv.org/abs/1702.04125.