Self-supervised Video Prediction

Lab Vision Systems: Learning Computer Vision on GPU´s

Mahpara Hyder Chowdhury and Pallab Das

Universität Bonn
s6machow@uni-bonn.de, Matrikelnummer: 3057801
s6padass@uni-bonn.de , Matrikelnummer: 3214085

Abstract. The ability to predict, anticipate and reason about future outcomes is a key component of intelligent decision-making systems. In light of the success of deep learning in computer vision, deep-learning-based video prediction emerged as a promising research direction. Defined as a self-supervised learning task, video prediction represents a suitable framework for representation learning, as it demonstrated potential capabilities for extracting meaningful representations of the underlying patterns in natural videos. Motivated from other video prediction tasks, we propose a model which generates future video frames taking a sequence of past frames. We train our model with UCF101 data set which is very large and diverse. Next, we classify the actions. Finally, we measure the accuracy of the action classification on the test set.

1 Introduction

In recent years, video prediction has become very popular among researchers. Thus, many research groups in the deep learning and computer vision communities have done deep analysis on it. This video prediction task consists of providing a model with a sequence of past frames, and asking it to generate the next frames in the sequence (also referred to as the future frames). This is challenging, as the model needs to embed highly-structured rich internal representations of the world and its physical rules. Typically in video prediction machine learning-based models, ground-truth future frames are provided as targets. This is also referred to as unsupervised training, as no manually-labelled data are needed. Training a model to predict future video frames is beneficial for a number of applications. First of all, the learned internal representations can be used for extracting rich semantic features in both space and time, which can then be utilized for different supervised discriminative tasks such as action or activity recognition, semantic segmentation, etc. Also, the ability to predict (or imagine) the future is an important skill of humans, that it used for anticipating the consequence of actions in the real-world, thus allowing us to make decisions about which action to perform. Hence, these internal video representations may support robots in their action decision process.

We propose a model for generating future video frames efficiently. The proposed model is evaluated on a data set UCF101. This data set is an action

recognition data set of realistic action videos, collected from YouTube, having 101 action categories. UCF101 gives the largest diversity in terms of actions and with the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc, it is the most challenging data set to date.

2 Related Work

Deep learning visual perception is a new perception convolutional neural network (Diego Rodriguez and Behnke, 2019). With this, it is possible to capture very bright lighting condition and extremely low environment. This visual perception system can recognize soccer-related objects, including soccer ball, boundaries of the field, robots, line segments and goalposts through the usage of texture, shape, brightness and color information. In their work, a pre-trained Resnet-18 is chosen as encoder. Since ResNet was originally designed for recognition tasks, they removed the Global Average Pooling(GAP) and the fully connected layers in the model. To use location-dependent features, they have used newly proposed location dependent convolution layer (Azizi et al., 2018). In the architecture, each convoBlock consists of two convolution layers followed by batch-norm and ReLu activations. Also, instead of a convolution layer, they have used a location-dependent convolution in the last layer. Different losses were used for different network heads. For detection head, the mean squared error is employed. In the classification head, they have used pixel-wise Negative Log Likelihood. The output of the network is of lower resolution and has less spatial information than the input image. To account for this effect in the detection part, they have calculated sub-pixel level coordinates based on the center of mass of a detected contour. Using a unified network helped both detection and segmentation. They got reduce training time as they did progressive image resizing and transfer learning techniques.

Deep convolutional neural networks are used to address many computer vision problems, including video prediction. The task of video prediction requires analyzing the video frames, temporally and spatially, and constructing a model of how the environment evolves. Video Ladder Network (VLN) (Cricri et al., 2016) is being used for efficiently generating future video frames. VLN is a neural encoder-decoder model augmented at all layers by both recurrent and feed-forward lateral connections. At each layer, these connections form a lateral recurrent residual block, where the feed-forward connection represents a skip connection and the recurrent connection represents the residual. For the presence of recurrent connections, the decoder can exploit temporal summaries generated from all layers of the encoder. This way, the top layer is relieved from the pressure of modeling lower-level spatial and temporal details. Moreover, they have extended the basic version of VLN to incorporate ResNet-style residual blocks in the encoder and decoder, which help improving the prediction results. VLN is trained in self- supervised regime on the Moving MNIST dataset, achieving

competitive results while having very simple structure and providing fast inference.

Convolutional neural networks are spatially invariant, though, which prevents them from modeling location-dependent patterns. Diego Rodriguez and Behnke (2019) proposed location-biased convolutional layers to overcome this limitation. The effectiveness of location bias was evaluated on two architectures: Video Ladder Network (VLN) and Convolutional Predictive Gating Pyramid (Conv-PGP). The results indicated that encoding location-dependent features is crucial for the task of video prediction. Their proposed methods significantly outperform spatially invariant models.

3 Methods

We are using pre-trained Resnet18 and taking the intermediate results as input for ConvGRU, 1x1 Conv and Location Dependent Conv. Refinement module is taking the last output of Resnet18. For loss, we use DSSIM+L1+L2 loss for training.

3.1 Proposed Model

This model (Fig:1) is inspired by https://arxiv.org/abs/1612.01756, and: winner2019.pdf.

3.2 ResNet-18

To extract relevant features from input video clip we use a CNN architecture-ResNet-18 which has demonstrated its ability with higher accuracy and lower computational cost. Furthermore, we use pretrained ResNet-18 to leverage the transfer learning in our implementation.
We use $224 \times 224 \times 3$ as input clip and got rid of classification layer of ResNet-18 architecture as our main goal is not classifying images. As a result, we get $7 \times 7 \times 512$ feature map as final output from ResNet and it as input for decoder module. We also use all four intermediate feature maps obtained from each layer of ResNet with Convolutional Gated Recurrent Unit, Location Depended Bias, and 1×1 convolution and stack them up in decoder module for predicting frames.

3.3 Convolutional Gated Recurrent Unit

Gated Recurrent Unit belongs to recurrent neural network family. GRU is similar to LSTM but has much simpler architecture which leads to less computational costs.

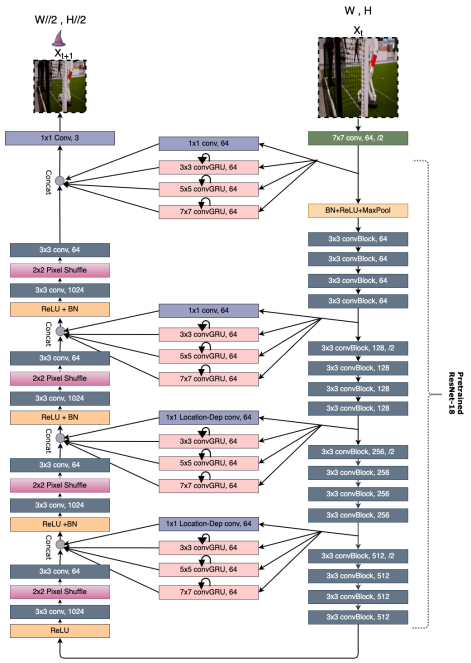

Fig. 1.

ConvGRU is combination of convolution, and GRU and dot products in gated architecture is replaced by convolutions.

$$z_t = \sigma(x_t * w_{xz} + h_{t-1} * w_{hz} + b_z), \tag{1}$$

$$r_t = \sigma(x_t * w_{rx} + h_{t-1} * w_{hr} + b_r), \tag{2}$$

$$\tilde{h_t} = tanh(x_t * w_{xh} + r_t \odot h_{t-1} * w_{h\tilde{h}} + b_{\tilde{h}}), \tag{3}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h_t} \tag{4}$$

where operators $\odot$ and $*$ represents element-wise multiplication and convolution operations respectively, σ is sigmoid activation fuunction and w and b's are parameters.
Our ConvGru takes in extracted feature maps from each layer of ResNet and previous state h_{t-1} to compute the new state h_t. The states and gates in ConvGRU architecture incorporates spatio-temporal advancement of objects in image sequences.

3.4 Location Dependent Bias

Intuitively, CNNs are invariant to locations of objects which is an advantage for image recognition but a disadvantage for video prediction as objects present in frames tend to change location over (consecutive) frames and time. To learn

these location dependent features extracted by Resnet, we introduced location dependency in the model. We concatenate the results with ConvGru results and utilized in decoder to obtain (high quality) prediction frames.

3.5 1x1 Convolution

We use a convolution with a 1×1 filter to work on low-level features extracted from ResNet and concatenate with ConvGRU output and decoder output while forecasting video frames. 1×1 convolution allows us to introduce non-linearity and retain the height and width of feature maps which helps to achieve better spatial dimension for our predicted output. Additionally, this convolution operation preserves the salient features while reducing dimensions of feature maps.

Our decoder module consists of four blocks of two convolutional layer with filter size 3×3 and a pixel shuffle layer with upscale factor 2×2. The output of decoder module is stacked with ConvGRU results and location dependent or 1×1 convolution outputs and feed it to next concurrent layer of decoder module. We also use batch normalization to stabilize and accelerate the training. We introduced pixel shuffle with the aim to produce high-resolution images from low resolution ones.

4 Results

We use a stochastic optimization algorithm Adam (Kingma and Ba, 2015) as optimizer for our training with learning rate of 0.0001. We trained our model in Google Colabratory with free gpus. To calculate losses, we use L1 loss, L2 loss and Difference of Structural Similarity(DSSIM) loss functions throughout our experiment. We evaluated our model on UCF-101 dataset with weighted sum of these loss functions. Table 1 highlights the values of cost functions on overall test dataset where L1 loss is lower than L2 loss. The weighted gain is obtained by multiplying L1 loss, L2 loss and DSSIM with $1, 0.5,$ and 2 respectively. Figure 2 depicts weighted sum over all the losses on training dataset at every epoch during our training phase. It clearly shows that our model is learning quite efficiently and converging smoothly.

Dataset	Cost Functions		
	L1	L2	DSSIM
UCF-101	0.9134	1.178	0.4354

Table 1. Average value of different image quality metrics on the test dataset. The value of DSSIM obtained by averaging three color channels.(Lower is better for all of them)

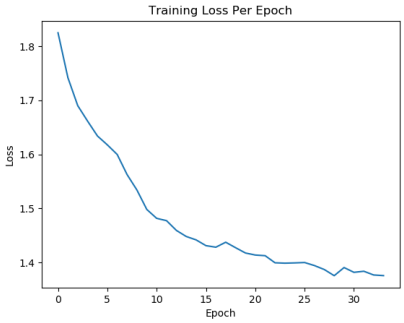

Fig. 2. Loss on training dataset per epoch

5 Conclusion

Video prediction is a promising avenue for the self-supervised learning of rich spatiotemporal correlations to provide prediction capabilities to existing intelligent decision-making systems. Our proposed model is a novel architecture for generating future video frames, conditioned on past frames. We took three seed frames and predicted the next three frames. After training the video prediction model, we added two linear layers on top of the hidden states from Conv-GRU, and classified the actions. We forwarded the full video to the model and saved the Conv-GRU's internal hidden states for that video. Finally, we classified the video's action using the save hidden states with an MLP with one hidden layer. While great strides have been made, we believe there is still room for improvement in video prediction using deep learning techniques.

References

Azizi, Niloofar et al. (2018). *Location Dependency in Video Prediction.*

Cricri, Francesco et al. (2016). "Video Ladder Networks". In: *CoRR*.

Diego Rodriguez Hafez Farazi, Grzegorz Ficht Dmytro Pavlichenko André Brandenburger Mojtaba Hosseini Oleg Kosenko Michael Schreiber Marcel Missura and Sven Behnke (2019). "RoboCup 2019 AdultSize Winner NimbRo:Deep Learning Perception, In-Walk Kick, Push Recovery, and Team Play Capabilities". In: *RoboCup 2019, Robot World Cup XXIII*, pp. 631–645.

Kingma, Diederik P. and Jimmy Ba (May 2015). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations* (San Diego, CA, USA). Ed. by Yoshua Bengio and Yann LeCun. URL: `http://arxiv.org/abs/1412.6980`.