PREDICTING VIDEO-FRAMES USING ENCODER-CONVLSTM COMBINATION

Subham Mukherjee[1], Spandan Ghosh[1], Souvik Ghosh [1], Pradeep Kumar[2], Partha Pratim Roy[2]*

[1]Institute of Engineering & Management, India [2]Indian Institute of Technology Roorkee, India

ABSTRACT

Video generation is an active field of research. With the rise in the amount of available data and economically available processing power in the form of GPUs, deep Learning has been a go-to solution for many real life problems and similarly it is often attempted to solve the problem of video generation using deep learning. Predicting the next set of frames for a given set of frames in a video has seldom been taken up. Each video is composed of a consecutive closely related frames of images. If we consider these frames, the frame in each time-step seems to be related to the frames in the preceding time-steps. Therefore, we have both spatial and temporal data available from any set of consecutive frames in a video. Learning some sort of representation of the images that *encodes* the spatial data of the images (frames) can be combined with learning *how these representations of a particular time-step is related with the next few time-steps* is made possible, then prediction of the next few frames for a given set of frames is made possible. Our aim is to propose a simple yet effective model that can achieve this goal.

Index Terms— Video Generation; Auto-encoder; ConvLSTM; Deep Learning

1. INTRODUCTION

With the rise of deep learning era more challenging tasks in computer vision can be solved. We propose a methodology to solve video generation task (e.g. future prediction) by using a state of the art technique that maintains the object spatial and temporal features. We know that deep convolutional neural network (CNN), works really well on image and video data. However, they cannot be solely used to generate videos as they miss out the temporal features of the data. To maintain the temporal features a sequence model is required. In recent times, as online data available and with the advancement of the Internet a large number of high quality unlabeled videos are available for free online apart from various labeled datasets. These can be acquired easily and can be considered to be of high quality because of the high degree of coherence available in real-life videos that are now present online.

We aim to differ from the popular approaches in video generation by using Generative Adversarial Networks(GANS) [1] and to use deep feature Consistent Variational Autoencoders by Hou et. al. [2] followed by sequence models using Conv-LSTMs as they are proven to better preserve the spatial data along with learning the temporal relationships essentially involved in the individual frames. Video generation is an active research area in the field of computer vision and deep learning. Some significant approaches deserving a special mention include [3, 4, 5, 6, 7, 8]. Long Short Term Memory (LSTM's) and the sequence-to-sequence model [9] have shown significant results in these applications [10, 11, 12, 13]. A common piece in these works is the use of a convolutional neural network (CNN) which encodes and decodes each frame and it's connected to a sequence-to-sequence model to predict the future frames.

Deep generative models are used for various state-of-the-art techniques. The two most popular generative models are he Variational Autoencoder (VAE) [14] and the Generative Adversarial Network (GAN) [1]. There are several GAN frameworks proposed for video generation. There was an attempt made by seperating scene and dynamic content [7]. Tulyakov et al. [15] used a RNN model for video generation into a GAN-based framework. This model was able to construct a video simply by pushing random noise into the RNN model. To better address the general video generation and to find a better and more elegant solution to generate the next few frames given a sequence of frames.We need to understand how pixels change from frame to frame to generate a full temporal object action. We note that there is generally a higher level of uncertainty in the exact movement between frames of moving objects. Intuitively we may tackle these problems by a mental reconstruction of these objects and understanding how their positions and orientations vary with respect to time (in our case, frames).

In this work, we understand the problem of learning how scenes transform with time. The solution of this problem will lead to better predictive models for computer vision. We lean this by using a large amount of unlabeled video. Unlabeled video has the advantage that it can be economically acquired at massive scales yet contains rich temporal signals 'for free' because frames are temporally coherent. We aim to present a simple and intuitive method to predict the next few frames in a video. To achieve this, we need to understand the images individually and then a relationship among these images in the time scale must be obtained. Our goal is to capture spa-

*Corresponding Author

tial and temporal knowledge contained in large amounts in the unlabeled videos and to predict and generate the next few frames which have fairly realistic dynamics and motions. For this, a variational autoencoder is used which maintains the spatial knowledge and a convolution LSTM which maintains the temporal knowledge.

2. PROPOSED MODEL

We suggest a two phase method of training the network in order to learn both the spatial and temporal data. A flow diagram of the proposed model is depicted in Fig. 1, where we take our entire dataset of videos and split it into sets of frames, and then we shuffle these frames. We suggest the use of deep feature Consistent Variational Autoencoders for learning the representation space for the images. This provides crisper and sharper representations than those formed by training a Convolutional autoencoder using MSE. After end to end training if this autoencoder, we then use the encoder and feed a series of representations of consecutive frames of the video to a Convolutional LSTM which learns to map a sequence of image representations to another sequence of image representations, thereby predicting the video frames in the future time-steps.

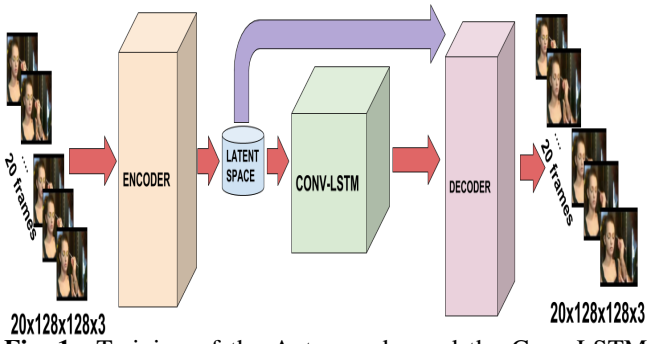

Fig. 1. Training of the Autoencoder and the Conv-LSTM. The purple arrow denotes the first training phase of the autoencoder and the red arrow denotes the final training phase with the Conv-LSTM.

2.1. The Autoencoder

Variational autoencoders were introduced by Pu, Yunchen and Gan et. al. [16] We however prefer the Deep Feature Consistent Variational autoencoder as was proposed by Hou, Xianxu and Shen et. al. [2] With the help of the VAE we encode an image x to a latent vector denoted by $z = Encoder(x) \sim q(z|x)$ with an encoder network, and then while training, use a decoder network is used to decode the latent vector z back to an image that will be as similar as the original image $\bar{x} = Decoder(z) \sim p(x|z)$. Thus, we can ensure that we have learnt an accurate representation space for our images by maximizing the marginal log-likelihood of each observation (pixel) in x. The VAE reconstruction loss $\mathcal{L}_{rec}$ is the negative expected log-likelihood of the observations in x.VAEs can further control the distribution of the latent vector z, which has characteristic of being independent unit Gaussian random variables, i.e., $z \sim \mathcal{N}(0, I)$. The difference between the distribution of $q(z|x)$ and the distribution of a Gaussian distribution (called KL Divergence) can be quantified and minimized by gradient descent algorithm. Therefore, VAE models can be trained by optimizing the sum of the reconstruction loss ($\mathcal{L}_{rec}$) and KL divergence loss ($\mathcal{L}_{kl}$) by gradient descent. We aim to use the autoencoder as a feature extractor. The auto-encoder was trained end to end in order to learn the features to be represented as a latent space z. This autoencoder is time-distributed and the output of its encoder is fed as an input for a Sequence Model where we have used a ConvLSTM in order to learn the relationships between the timeframes while preserving the spatial data. This then passes through the decoder which then gives us the next few frames for the given frames to the encoder. We first train this autoencoder end to end to find an optimum latent space z.

$$\mathcal{L}_{rec} = -\mathbb{E}_{q(z|x)}[\log p(x|z)] \tag{1}$$

$$\mathcal{L}_{kl} = D_{kl}(q(z|x)||p(z)) \tag{2}$$

$$\mathcal{L}_{vae} = \mathcal{L}_{rec} + \mathcal{L}_{kl} \tag{3}$$

2.2. Convolutional LSTM

This learned representation space is of great value to this problem. After saving weights, the encoder is separated from the decoder. Then, a set of frames of a video, pass them all through the encoder to get $z = Encoder(x)$ and then we take these encoded vectors and pass them through a Conv-LSTM and train it to predict the next few frames. The internal operation of the ConvLSTM is shown in Fig. 2 and Fig. 3. This approach was introduced by Xingjian et al. [17] for Precipitation Nowcasting.

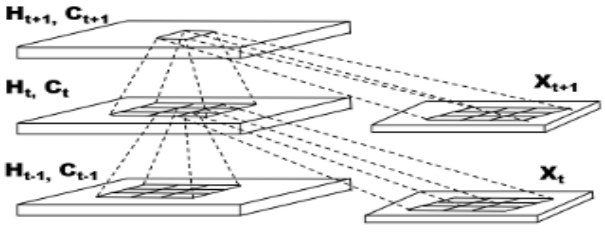

Fig. 2. Inner structure of ConvLSTM.

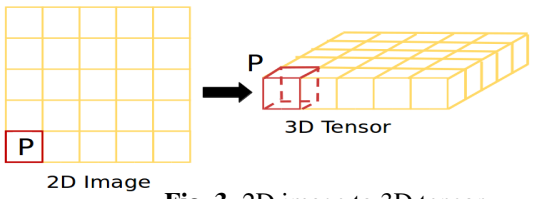

Fig. 3. 2D image to 3D tensor.

2.3. Our Approach

We have a latent space z which encodes the previous frames of a video and intend to predict the next few frames. From the machine learning perspective, this problem can be regarded as a spatio-temporal sequence forecasting problem. Let us assume that our latent space z is an $M \times N$ grid which consists of M rows and N columns. Inside each cell in the grid, there are P measurements which vary over time. For each time-step, the frames can be represented by a tensor $\mathcal{X} \in \mathbf{R}^{P \times M \times N}$, where $\mathbf{R}$ denotes the domain of the observed features in the representation.Operating over several periods will get a sequence of tensors $\hat{\mathcal{X}}_1, \hat{\mathcal{X}}_2, \ldots, \hat{\mathcal{X}}_t$. We treat video prediction as a spatio-temporal sequence forecasting problem. The problem lies in predicting the most likely length-K sequence in the future given the previous J observations which include the current one. The ConvLSTM is governed by the equations 4.

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \\
o_t &= \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o) \\
\mathcal{H}_t &= o_t \circ \tanh(\mathcal{C}_t)
\end{aligned}
\tag{4}
$$

3. EXPERIMENTAL RESULTS

3.1. Data

The evaluate of our model is done on KTH actions [18] and UCF101 dataset [19]. The KTH action dataset is a video database containing six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in four different scenarios. UCF101 is an action recognition data set of realistic action videos, collected from YouTube, having 101 action categories. UCF101 consists of 13320 videos from 101 action categories. UCF101 gives the largest diversity in terms of actions and with the presence of large variations.

3.2. The Running model

We have an encoder that for a set of frames gives us a latent representation space z which is used by the Conv-LSTM to generate the latent space $z'(say)$ of another set of frames. These are then passed through the decoder to get the next few frames from the model. The encoder and decoder network are trained first to learn the spatial features of the data. After that the Conv-LSTM is inserted between the encoder and the decoder. The weights of encoder-decoder are freezed and then the Conv-LSTM is then trained. At the final stage of training the whole network along with the encoder-decoder is trained. The datasets mentioned above were collected and preporocessed by first extracting frames out of these videos and resizing them to (128x128) and normalizing them. These images were shuffled and fed to autoencoder for end-to-end pre-training. It was observed that if these images are not shuffled then several similar images present at the end of the training process bias the autoencoder towards the last batches of images and it does not generalize well to the entire dataset of extracted images from videos. The Encoder- decoder architecture is as shown in Fig. 4

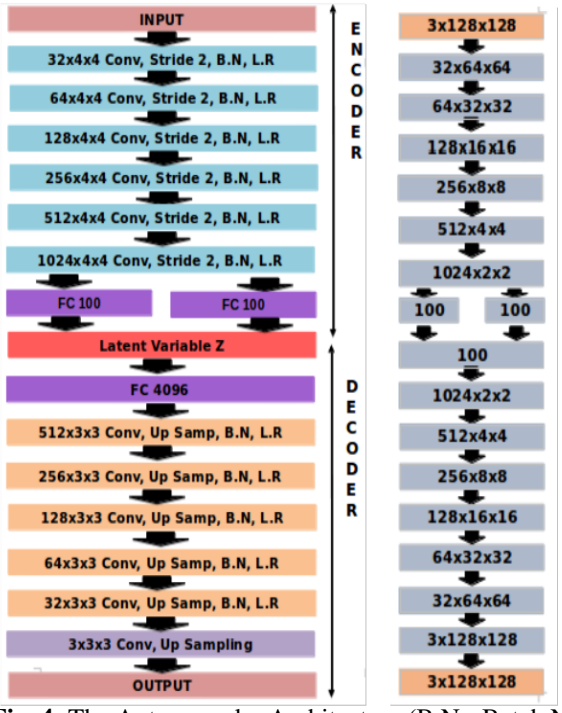

Fig. 4. The Auto-encoder Architecture (B.N.: Batch Normalization L.R: Leaky ReLU).

Subsequent to this end to end training the weights of the autoencoder were freezed and then we proceeded to train the Convolutional LSTM to learn the temporal relationships while preserving the spatial data present in the Latent space z learned by the autoencoder. The ConvLSTM layer not only preserves the advantages of FC-LSTM but is also suitable for spatiotemporal data due to its inherent convolutional structure. By incorporating ConvLSTM into the encoding-forecasting structure, the spatial data has been preserved while learning the temporal relationships. The first 20 frames

from the videos were taken as input set and the next 20 frames were taken as output frames. We assigned a completely white frame as the 'START' indicator and a completely black frame as an EOL (end of Line or in this case, the end of the sequence). Following this the Conv-LSTM was trained using a kernel size of 3x3 and 256 filters with the help of the already trained autoencoder with the RMSProp optimizer. The Loss functions converged as shown in Fig. 5 and Fig. 6. Subsequently the entire model was trained once end to end for fine-tuning the weights with a very low learning rate (1e-5).

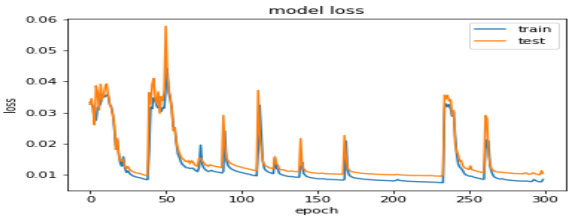

Fig. 5. DFC-VAE Loss Function

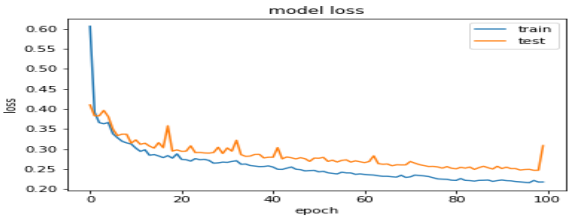

Fig. 6. CONV-LSTM Network.

3.3. Evaluation

The DFC-VAE loss function behaved smoothly while converging. However the Conv-Lstm loss function had a lot of spikes during the initial phase of training. Sequence models suffer from vanishing and exploding gradients. For our experiment to prevent exploding gradient we have used gradient clipping with a clip value of 1 and for vanishing gradient we have applied RMS-Prop optimization. For quantitative analysis, we chose the predicted frames and take the same frames from the ground truth video. Then these frames were compared as images and the parameters were averaged out to analyze. PSNR and SSIM have been used for quantitative analysis of the output. Five predicted frames from two distinct videos have been compared to the ground truth frames. The mean PSNR and SSIM are also provided for reference and provided in the table 1 and Fig. 7 and Fig. 8. Samples of predicted frames from the KTH dataset are also shown in Fig. 9.

3.4. Conclusion

Our experiment was successful to produce output which maintained both the spatial as well temporal features properly. Though it was noticed that the few frames that were predicted at the last were a bit blurry. The future aspect of our experiment is to find a better method that will improve the spatial quality of the last few frames. The loss function used in our experiment also has a scope of improvement.

Table 1. PSNR and SSIM Values

Predicted Frame	PSNR	SSIM
1	31.81	0.868
2	31.82	0.869
3	31.845	0.870
4	31.88	0.873
5	31.90	0.875

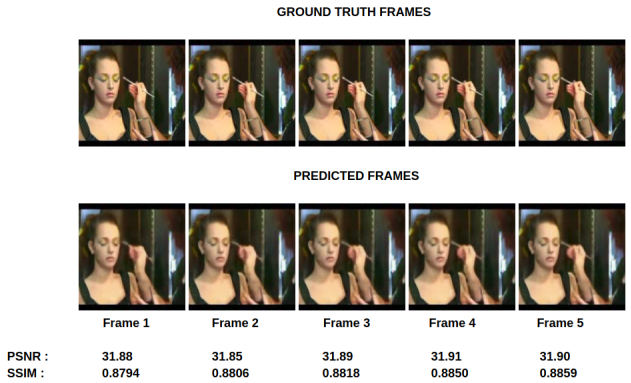

Fig. 7. Evaluation on a sample (UCF101).

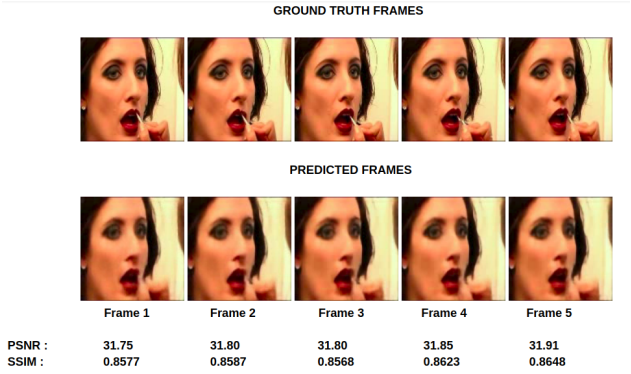

Fig. 8. Evaluation on a sample (UCF101)

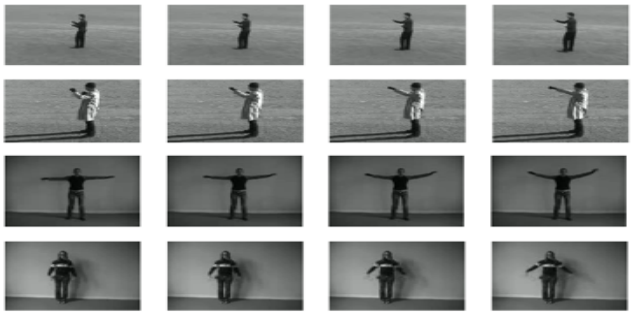

Fig. 9. Output of Our Experiment (KTH).

4. REFERENCES

[1] Ian Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[2] Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu, "Deep feature consistent variational autoencoder," in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*. IEEE, 2017, pp. 1133–1141.

[3] Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P Xing, "Dual motion gan for future-flow embedded video prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1744–1752.

[4] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala, "Video frame synthesis using deep voxel flow," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4463–4471.

[5] Tanya Marwah, Gaurav Mittal, and Vineeth N Balasubramanian, "Attentive semantic video generation using captions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1426–1434.

[6] Lei Chen, Jiwen Lu, Zhanjie Song, and Jie Zhou, "Part-activated deep reinforcement learning for action prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 421–436.

[7] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba, "Generating videos with scene dynamics," in *Advances In Neural Information Processing Systems*, 2016, pp. 613–621.

[8] Emily L Denton et al., "Unsupervised learning of disentangled representations from video," in *Advances in Neural Information Processing Systems*, 2017, pp. 4414–4423.

[9] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[10] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee, "Decomposing motion and content for natural video sequence prediction," *arXiv preprint arXiv:1706.08033*, 2017.

[11] Joost Van Amersfoort, Anitha Kannan, Marc'Aurelio Ranzato, Arthur Szlam, Du Tran, and Soumith Chintala, "Transformation-based models of video sequences," *arXiv preprint arXiv:1701.08435*, 2017.

[12] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C Cobo, Florian Stimberg, et al., "Parallel wavenet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[13] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert, "The pose knows: Video forecasting by generating pose futures," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 3352–3361.

[14] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[15] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz, "Mocogan: Decomposing motion and content for video generation," *arXiv preprint arXiv:1707.04993*, 2017.

[16] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin, "Variational autoencoder for deep learning of images, labels and captions," in *Advances in neural information processing systems*, 2016, pp. 2352–2360.

[17] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[18] Christian Schuldt, Ivan Laptev, and Barbara Caputo, "Recognizing human actions: a local svm approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. IEEE, 2004, vol. 3, pp. 32–36.

[19] 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.