3D VAE Video Prediction Model with Kullback Leibler Loss Enhancement

Zahraa Al Mokhtar
Department of Engineering, Computer Engineering
University of Mosul, Iraq
zahraatalal84@gmail.com

Shefa Dawwd
Department of Engineering, Computer Engineering
University of Mosul, Iraq
shefa.dawwd@uomosul.edu.iq

Abstract: *The Video Prediction (VP) models adopted many techniques to build suitable structures to extract the spatiotemporal features and predict the future frame. The VP techniques extracted the spatial and temporal features in separated models and then fused both features to generate the future frame. However, these architectures suffered from the design complexity and time for prediction required. So, many efforts introduced VP based on decreasing design complexity and producing good results. This study produces the VP model based on a Three-Dimensional Variational Auto Encoder (3D VAE). The proposed model builds all layers depending on 3D convolutional layers. This leads to better extraction of spatiotemporal information and decreases the design complexity. Second, the Kullback Leibler Loss (KL Loss) is enhanced by a 3D sampling stage which allows to calculation of the 3D latent loss. This helps to extract the better and proper spatiotemporal latent variable from the 3D Encoder. The 3D sampling represents a good regularizer in the model. The proposed model outperforms in terms of SNR=34.8673, SSIM= 0.9616 which applied to Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) and Caltech pedestrian datasets and records 5.2 M parameters*.

Keywords: *3D sampling stage, kullback leibler loss, temporal 3D sampling stage, three dimensional RNN layers, variational autoencoder*.

Received April 4, 2024; accepted September 8, 2024
https://doi.org/10.34028/iajit/21/5/9

1. Introduction

The prediction of future frame(s) or events is a critical task of intelligent true decision-making in real-time applications which are inspired by the operation of the human brain [25]. So, the researchers faced a big challenge to build Deep Learning (DL) models that simulate human capabilities to create a proper decision effortlessly and quickly [7]. Generally, the most important DL technique applied in a Video Prediction (VP) model is the AE structures which are entirely built by the classical Convolutional Neural Network (CNN) Two-Dimensional Convolution (2D CONV), Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM), and Convolutional Long Short Term Memory (CONVLSTM) layers [7, 19].

The classical Auto-Encoder (AE) models are used to extract low-level features in an unsupervised fashion from high-dimensional input [11]. The basic concept of AE models is typically reducing the dimensionality of the input data and producing low-level features in the encoder part, then decomposing the compressed features to get the predicted results or frame(s) in the decoder part [4].

However, most 2D AE models suffer from blurry prediction results due to the inconsistency between spatial and temporal features and the uncertainty probability of data [23, 27]. So, many algorithms are proposed to enhance the prediction performance by adding many layers in the encoder and decoder parts [7], building the intermediate block between the encoder and the decoder, and proposing another version of AE like Variational Autoencoder (VAE) and so on [3].

Desai *et al*. [7] proposed a VP model based on the CONVLSTM encoder-CONVLSTM decoder. Wang *et al*. [33] proposed a VP model that combined the 3D-LSTM with 3D CONV based on the AE. While Lotter *et al*. [20] suggested a Predictive Neural Network Model (PredNet) which combined the ConvLSTM with a predictive coding concept to predict the next frames by creating a local prediction in each layer. Villegas *et al*. [32] introduced a 2D AE model with a CONVLSTM as the bottleneck stage to predict the next frame at the pixel level. While Straka *et al*. [30] suggested AE with Predictive Coding Net (PreCNet) which is applied as an estimator block between the encoder and decoder parts. Ye *et al*. [37] presented an AE model with an intermediate Neural Process (NP) block that maps spatiotemporal input coordinates to produce each pixel value of the output. Gao *et al*. [9] proposed a simple spatial-temporal features translator between the encoder and the decoder part to enhance the blurry prediction.

On the other side, many researchers began to combine the AE model with distribution probability in an explicit manner to build a new version called VAE and solve the uncertainty prediction problem, Castrejon *et al*. [3] applied a 2D-VAE with the hierarchy of latent variables. This creates groups of flexible distributions to get a

better probability of future frame(s). Lu *et al*. [22] suggested a new sequential model of VAE based on CONVLSTM to assess the abnormal behaviour in the next frame. Pan *et al*. [26] proposed a new model based on conditional VAE (cVAE) which applied to the semantic label. This model contains two steps. The first stage generates the starting frame based on the semantic label, while the second stage uses the image-to-video (img2vid) network to get a video sequence from the initial frame. Wu *et al*. [35] proposed a VAE based on a Greedy Hierarchical (GHVAE) which learns high-fidelity next frame predictions by training each level of the autoencoder. Finally, Razali and Fernando [29] proposed the dual model cVAE based on ResNet-18 and 3D CONV layers to enhance the predicted frame.

All the above papers enhanced the predicted result but raised the complexity of models by increasing the number of encoder and decoder layers or adding the intermediate CNN blocks. These models failed to strike a balance between their performance and the number of parameters which reflected at the time of prediction. In this work, a generative semi-supervised 3D VAE model is introduced to overcome the inconsistency in spatial-temporal features by using the 3D CONV layers and 3D sampler. The proposed 3D sampler captures the distribution probability in spatial and time-space simultaneously. This increases the consistency between context and time features and produces better prediction results.

The main contribution of the proposed work can be summarized in two points: First, we present a new structure of the VAE model based on 3D CONV and 3D CONVLSTM layers on the entire architecture and show their effectiveness for future frame prediction. These 3D layers play an important role in capturing the spatial and temporal features simultaneously to decrease the blurry prediction and increase the prediction performance. Second, we extend the 2D KL divergence by introducing a 3D KL_loss based on 3D CONVLSTM. This 3D-distributed probability regularizes the latent space and increases the consistency between the spatial and temporal features. Finally, we show through the experiments of the proposed model that each contribution enhances the model and their combination permits it to outperform present state-of-the-art VP models

2. The Classical Variational Autoencoder

The classical VAE is a DL architecture consisting of three networks based on 2D CONV layers [14, 36] as shown in Figure 1. The Encoder-Decoder parts are trained to minimize the reconstruction error between the target or Ground Truth (GT) data and generated data from the decoder. However, The VAE model has a great regularization of the latent space compared with the classical AE model. The distribution probability density is explicitly applied to low-level extracted features. The model is designed as follows: First, the input data or sequence of frames is encoded as a latent space of low-level features. Second, the latent space variables are sampled based on the probability distribution. Third, the sampled point is decomposed in the decoder model. finally, the reconstruction error is calculated as back propagated error through the model layers [31]. The VAE operations can be written as follows [34]:

$$z = Encoder\ (X) = \ q(z|x) \tag{1}$$

$$O = Decoder\ (z) = p(o|z) \tag{2}$$

Where X={x1, x2,⋯, xt}, xt ∈ R. xt are input sequence frames. O={ ot+1, ot+2, ..., ot+M }. It is predicted frames.

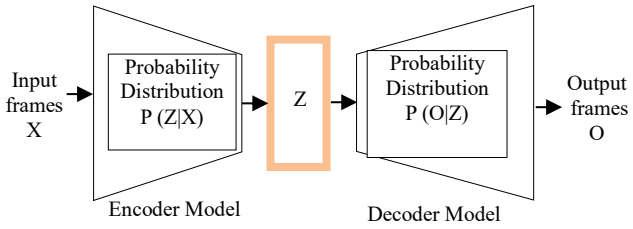

Figure 1. General design of VAE model.

3. The Proposed Model

The 3D CONV layers are used in the encoder and decoder parts. Logically, the VP models based on 2D CONV layers may not be able to process the spatial and temporal space simultaneously. Each frame will be handled in the spatial space independently. While the 3D CONV layers can extract the spatial and temporal features simultaneously. In practice, the 3D CONV operations are performed better [6]. The model consists of three components: The 3D encoder, the 3D Z-sampling, and the 3D CONV decoder parts as shown in Figure 2. For simplicity, batch normalization, the Max-pooling, and activation layers are not displayed in the figure.

The 3D Encoder consists of two stages: The first stage is performed as four 3D CNN layers with ReLU activation, Batch normalization, and Maxpooling3D layers. The encoder part extracts the Spatio-Temporal features simultaneously. However, every five consecutive frames are grouped as one block input of 256x256 RGB frame. So, the input sequence is applied as (5, 256, 256, 3). The output of each 3D CNN layer is down-sampled by two to reduce the spatial dimensionality of resulted features. Thus, the features extracted from the fourth layer are represented as (5, 32, 32, 128). The second stage of the 3D Encoder consists of three 3DCNN layers which compresses the time-space features while reserving on the dimensionality of the x, and y spaces as shown in Table 1.

The prior VAE architectures apply the 3D CNN layers in the encoder and the decoder parts and 2D CONVLSTM in the sampling part [3]. This leads to calculating the 2D kl_loss in the spatial domain only

[16]. However, the 3D VAE model applies the 3D CONVLSTM in the sampling stage to extract the distribution probability in the x, y and time domains. The 3D CONVLSTM consists of three gate units, and the operation formulas are identical to those in 2D CONVLSTM. However, the basic difference from 2D CONVLSTM is that the whole 3-D data is processed as the input of each memory cell in 3D CONVLSTM [15]. the formula of the 3D CONVLSTM cell can be described as [15]:

$$i_t = \sigma(W_{xi} \circledast X_t + W_{hi} \circledast H_{t-1} + W_{ci} \circledast C_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{xf} \circledast X_t + W_{hf} \circledast H_{t-1} + W_{cf} \circledast C_{t-1} + b_f) \quad (4)$$

$$\tilde{C}_t = tanh(W_{xc} \circledast X_t + W_{hc} \circledast H_{t-1} + b_c) \quad (5)$$

$$C_t = f_t . C_{t-1} + i_t * \tilde{C}_t \quad (6)$$

$$O_t = \sigma(W_{xo} \circledast X_t + W_{ho} \circledast H_{t-1} + W_{co} \circledast C_{t-1} + b_o) \quad (7)$$

$$H_t = O_t . \tanh(C_t) \quad (8)$$

where:

X_t is the input state. C_t and C_{t-1} are the state units. H_t, H_{t-1} are the output of 3D CONVLSTM. I_t, f_t, O_t are the input, forget, and output gate units of 3D CONVLSTM. W.i, W.f, W.o: are 3D CONV filters. The 3D CONVLSTM allows extracting the context and time-space of the latent variables at the same time by applying the three-dimensional multiplier in each gate. As an example in input gate, suppose that $X_t \in R^{TxWxH}$ and $W_{xi} \in R^{k1xk2xd}$ where T, H, W are the time, height, and width of frames or input data, and k1, k2, d are the dimensions of the filter. So the convolutional operation between X_t and W_{xi} can be described as [15]:

$$W_{xi} \circledast X_t = \sum_{l=0}^{k1} \sum_{m=0}^{k2} \sum_{n=0}^{d} W_{xi}^{(m,n)} * X_t^{(l,m,n)} \quad (9)$$

Where W_{xi} is the 3D CONV filters, X_t is the input state. The 3D sampling extracts the probability latent variable from spatial-temporal features, but the T-sampling introduces the probability variable based on temporal features by applying the transpose function to the output of 3D CONVLSTM. This helps to decrease the inconsistency of temporal features because the T-sampling calculates the distribution probability function based on temporal features. The (5, 32, 32, 128) features are changed to (32, 32, 5, 128). So, the T-sampling makes additional regularization based on the temporal features which increases the accuracy of the predicted frame.

The VAE can be represented as two players supporting each other to generate the best-predicted result based on the distribution function. The first player is the 3D Encoder part which is training to capture the best spatial temporal latent variable based on KL_loss as shown in Figure 2. The second player is the 3D Decoder part which changes the weights according to measure different types of metrics like Peak Signal to Noise Raito (RSNR), Structure Similarity Index Metric (SSIM), and Mean Square Error (MSE).

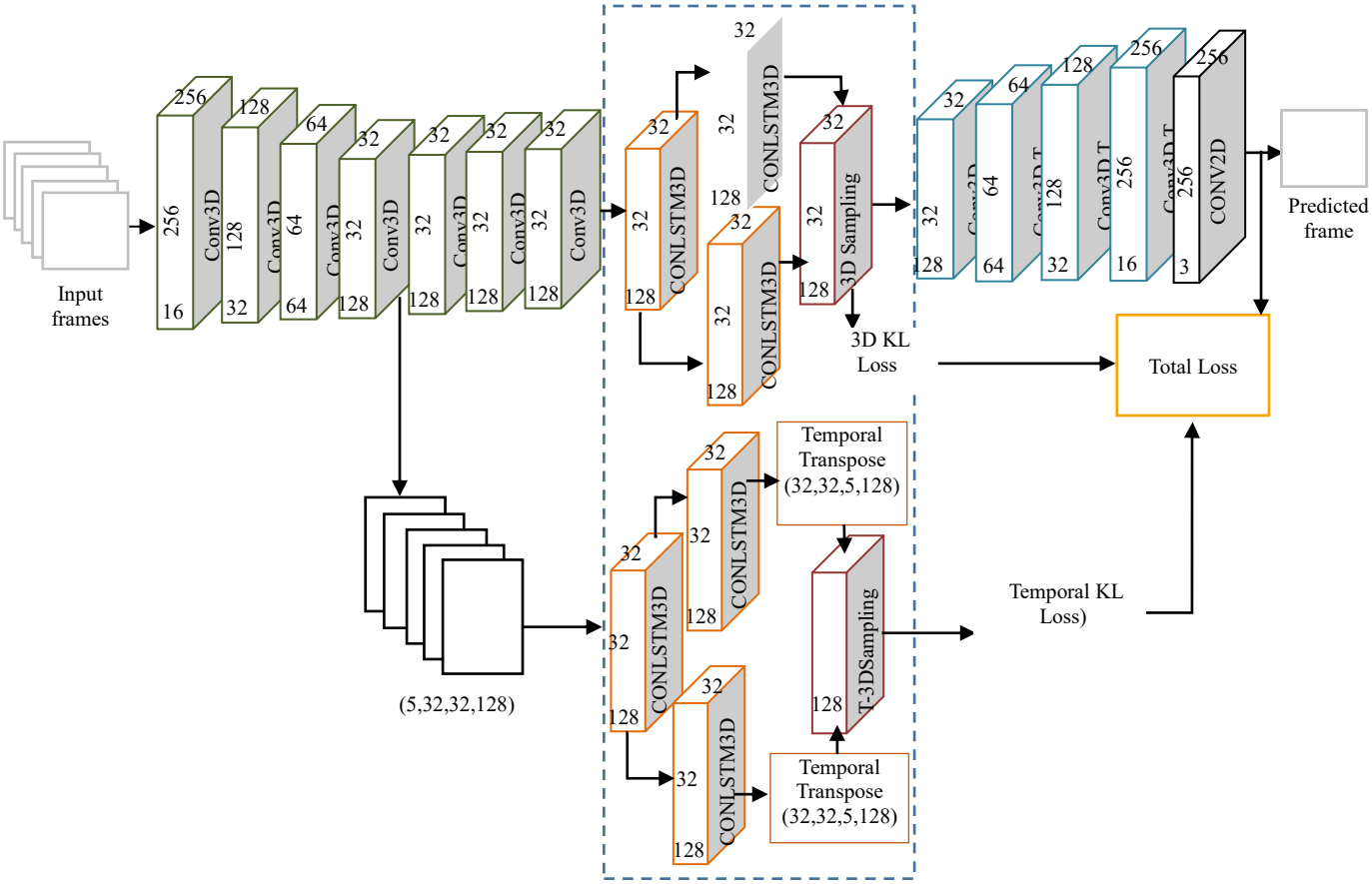

Figure 2. The proposed 3D-VAE model.

Table 1. Detail specification in each layer of the proposed model.

Layer type	Kernel size	Filter number	Stride	Description
CONV3D	(5,5,5)	16	(1,1,1)	Spatio-temporal in 3D Encoder
Maxpooling3D	(1,2,2)	------	(1,1,1)	
CONV3D	(5,5,5)	32	(1,1,1)	
Maxpooling3D	(1,2,2)	------	(1,1,1)	
CONV3D	(5,5,5)	64	(1,1,1)	
Maxpooling3D	(1,2,2)	------	(1,1,1)	
CONV3D	(5,5,5)	128	(1,1,1)	
CONV3D	(5,5,5)	128	(1,1,1)	Temporal compressed in 3D Encoder
Maxpooling3D	(2,1,1)	------	(1,1,1)	
CONV3D	(5,5,5)	128	(1,1,1)	
Maxpooling3D	(2,1,1)	------	(1,1,1)	
CONV3D	(5,5,5)	128	(1,1,1)	
Maxpooling3D	(2,1,1)	------	(1,1,1)	
CONLSTM 3D	(5,5,5)	128	(1,1,1)	
CONLSTM 3D	(5,5,5)	128	(1,1,1)	3D μ_ calculation
CONLSTM 3D	(5,5,5)	128	(1,1,1)	3D σ_ calculation
Sampling	3D –sampling			
CONLSTM 3D	(5,5,5)	128	(1,1,1)	
CONLSTM 3D	(5,5,5)	128	(1,1,1)	Tμ_ calculation
CONLSTM 3D	(5,5,5)	128	(1,1,1)	Tσ_ calculation
Sampling	Temporal-3D sampling			
CONV3D T	(5,5,5)	128	(1,2,2)	Reconstruction frames in 3D-decoder
CONV3D T	(5,5,5)	64	(1,2,2)	
CONV3D T	(5,5,5)	32	(1,2,2)	
CONV3D T	(5,5,5)	16	(1,2,2)	
CONV2D	(3,3)	3	(1,1)	

4. Evaluation Metrics

There are two types of loss functions which applied in the VAE model: The first type is the most popular metrics in image/video processing. These metrics determine the performance of the model by comparing the Predicted Frame (PF) with the GT. Generally, the most important metrics in VP applications are the MSE, PSNR, SSIM, etc., The PSNR is frequently employed as a means of signal reconstruction quality monitoring. The PSNR can be described as [21]:

$$PSNR(GT\ ,PF\ ) = 10\log\frac{255^2}{\sum_{i=0}^{N}(GT^i\ - PF^i\ )^2} \quad (10)$$

$$MSE(GT,PF) = \frac{1}{N}\sum_{i}^{N}\sum_{j}^{N}\ (GT^{i,j} - PF^{i,j})^2 \quad (11)$$

The SSIM is one of the best metrics that measures the mean of the frame based on gauge brightness, structural information, and variance factors. These three factors are more closely related to human perception. It is typically employed in applications that require the evaluation of image quality such as image super-resolution, image compression, and others. The SSIM can be described as [21]:

$$SSIM((\mathrm{GT},\mathrm{PF}) = \frac{(2\mu_{\mathrm{GT}}\ \mu_{\mathrm{PF}} + c1)(\ 2\ \sigma_{PFGT} +\ c2\ )}{(\mu_{\mathrm{GT}}{}^2\ {}_+\mu_{\mathrm{PF}}{}^2 + c1)(\ \sigma_{\mathrm{GT}}{}^2\ +\ \sigma_{\mathrm{PF}}{}^2 + c2\ )} \quad (12)$$

Where μ_{Y_1} is the mean of the GT frame. $\mu_{Y_2}$is the mean of the predicted frame.

σ_{Y_1} is the variance of the GT frame,σ_{Y_2} is the variance of the predicted frame.

$\sigma_{Y_2Y_1}$ is the covariance of GT, PF.

c1 and *c2* are described as [24]:

$$C1 = (\ k_1\mathrm{L})^2\ \ and\ C2 = (\ k_2\mathrm{L})^2 \quad (13)$$

C1 and *C2* are utilized to preserve the stability of the computational procedure.

L determines the dynamic range of each pixel value. K_1, K_2 are constant, in this approach, we choose k_1 =0.01 and k_2=0.03.

The values of SSIM are in the range of [-1, 1]. The SSIM can be indirectly calculated based on a sliding window operation on the original image which uses a Gaussian distribution convolution kernel. The average operation can be applied to the SSIM values of the image blocks. Generally, the size of the sliding window is set to 11×11, and the variance of the Gaussian distribution is 1.5.

The second type of loss function in VAE is Kullback Leibler divergence Loss (KL Loss). KL Loss calculates a dissimilarity between the Normal distribution N(0, 1) and the latent space distribution [1].

The classical KL loss can be described as [24]:

$$D_{KL}[q(z|x)||p(z)] = -\frac{1}{2}\left[\log\sigma^2\ + 1 - \sigma^2\ - \mu^2\ \right] \quad (14)$$

Where:

$$Z = \ \mu + \epsilon * \sigma$$
$$\sigma\ is\ the\ standard\ devation, \sigma\ \in R^z$$
$$\mu : is\ the\ mean\ value, \qquad \mu \in\ R^z$$
$$\epsilon\ is\ a\ Normal\ Distirution\ of\ Latent\ variables, \epsilon \in N(0, I)$$

The 3D KL-loss is extended based on 2D Kl_loss in Equation (14) to measure the divergence loss in context and time-space at the same time. The 3D KL loss can be described as:

$$D_{3DKL}[q(z|x)||p(z)] = -\frac{1}{2}\left[\sum_{t,i,j}\log 3D\sigma^2\ + 1 - \sum_{t,i,j} 3D\sigma^2 - \sum_{t,i,j} 3D\mu^2\right] \quad (15)$$

Where: *i*, *j* are spatial-space dimensional and *t* is the spatial temporal space dimensional.

The temporal-3D sampling regulates the temporal features by applying the transpose function to scan the features in the time domain for all values of the spatial index as described in Equation (16).

$$D_{TKL}[q(z|x)||p(z)] = -\frac{1}{2}\left[\sum_{i,j,t}\log T\sigma^2\ + 1 - \sum_{i,j,t} T\sigma^2 - \sum_{i,j,t} T\mu^2\right] \quad (16)$$

The reconstruction loss of the 3D VAE model is updated to include 3D KL_Loss in Equation (15), *T* KL_loss in Equation (16) and the classical MSE loss in Equation (11). The total loss can be written as follows:

$$TotalLoss_{vae} = 3DKL_{loss} + MSE_{loss} + TKL_{loss} \quad (17)$$

4. Experiments

4.1. Training Details

Keras and TensorFlow are used to implement the model

as Python code. The Adam optimizer is tuning at learning rate lr=0.0001; β1=0.9; β2=0.999. The Batch size=5 and epoch=70. The system is executed and applied on an NVIDIA RTX3060 GPU with 12 Giga Bytes memory. The training and testing operation is applied in an end-to-end way

4.2. Datasets

4.2.1. KITTI [10]

This is the most public standard dataset for computer vision models like VP, autonomous driving, and mobile robotics applications. It consists of 57 videos with 1392x512 RGB pixel resolution based on hours of traffic scripts applied with a different modality of sensors. This dataset included grey-scale stereo cameras, high-resolution RGB, and a 3D-laser scanner. The original dataset does not contain the semantic segmentation GT data. However, in 2015, the KITTI dataset was modified by adding a 200-frame for both instance and semantic segmentation in a pixel-level formula.

4.2.2. Cityscapes [5]

This dataset is very similar to the KITTI dataset and many papers used the KITTI and Cityscapes together. The Cityscapes introduces a large-scale database based on 50 videos with 2048x1024 RGB pixel resolution. This set was recorded in 50 different cities spending several months in good conditions weather, and daytime. It contains an instance label, semantic-label, stereo pairs of frames, and dense pixels for 30 categories grouped into 8 classes of urban street scenes. It is approximately composed of 5000 fine-explained images (1 frame in 30 seconds) and 20.000 annotated coarse ones (one frame every 20 seconds or 20 meters recorded by the car). It also consists of extra High-level data like outside temperature, vehicle sensors, and GPS tracks to serve a wide range of computer vision applications.

4.2.3. Caltech Pedestrian Dataset [8]

This data is introduced to detect pedestrians as a driving dataset. The bounding boxes are utilized to capture a pedestrian. It is approximately recorded at 10 hours of 640x480 with a frame rate of 30fps. The video series is extracted from a vehicle driving across uniform traffic in an urban environment. It contains 250k frames collected from 137 clips. The length of each clip is approximately one minute. The total number of pedestrians bounding boxes is 350.000, specifying two identical pedestrians.

5. Results and Discussion

5.1. Single Frame Prediction

The first step of any VP model choose a suitable number of input frames to capture the proper spatial and temporal features. Thus, the number of input frames are increasing gradually and the performance of the model is measured at each time as shown in Table 2. The values of PSNR, MSE, and SSIM are generally constant in 5 frames and above. So, the five input frames are a suitable number to get better performance with the minimum number of parameters. Second, the proper datasets like KITTI and cityscapes are applied to the proposed model in the training phase and measure the performance.

Table 2. Different numbers of input frames applied to the proposed model.

Input frames	MSE	PSNR	SSIM
3	0.009	29.905	0.881
4	0.00094	32.812	0. 91
5	0.000485	34.8673	0.9616
6	0.000485	34.8673	0.9616
7	0.000485	34.8673	0.9616
8	0.000485	34.8589	0.9616
9	0.000485	34.8673	0.9616
10	0.000485	34.8673	0.9616

The effects of the 3D sampling and Temporal-3D sampling are represented in Table 3. The 3D VAE model based on 3D sampling records a good result. However, some blurry predictions are due to the temporal latent probability variable not being captured very well. So, the Temporal-3D sampling helps to capture the proper temporal latent features and enhance the prediction performance.

Table 3. Comparison study of the proposed model.

3D Encoder	3D Decoder	3D sampling	Temporal-3D sampling	PSNR
√	√	X	X	31.59
√	√	X	√	32.2
√	√	√	X	33.3
√	√	√	√	34.86

The qualitative analysis of the next frame predicted looks good and the edges look smooth too compared to the GT as shown in Figure 3. We can see that the 3D VAE model is outperforming in MSE, PSNR, SSIM, and the number of parameters in comparison to the state-of-the-art models as shown in Table 4.

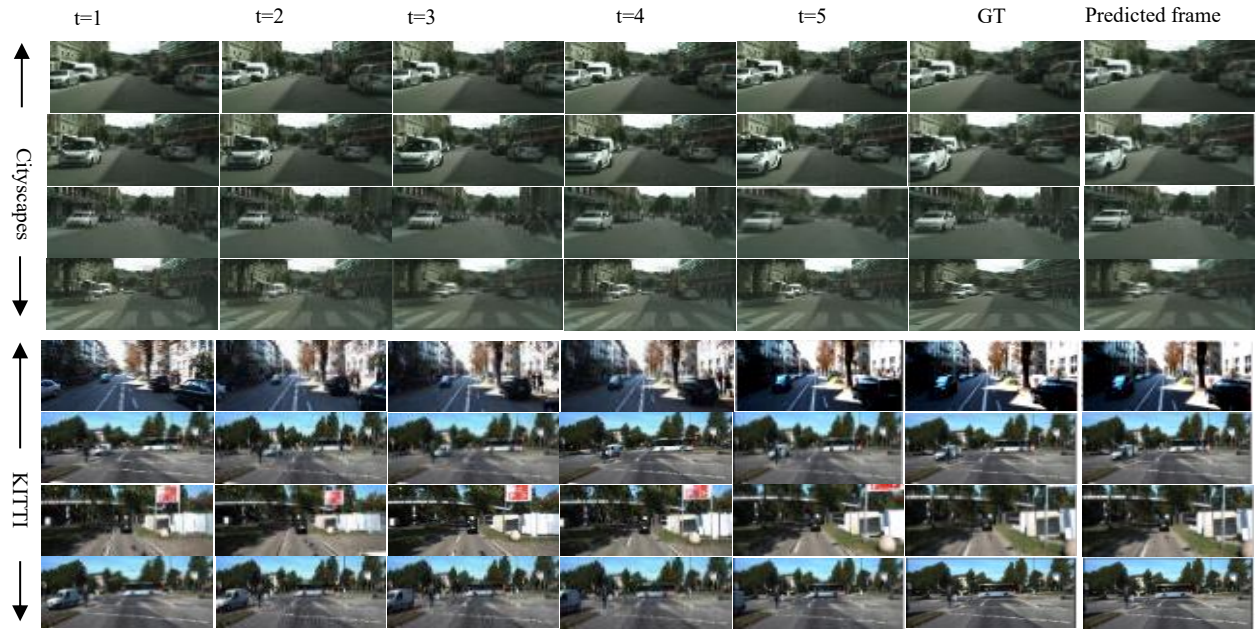

Figure 3. The qualitative analysis from Cityscapes and KITTI training datasets. The first five column contain actual sequence of input frames. The sixth column contains the GT. The final column describes the results.

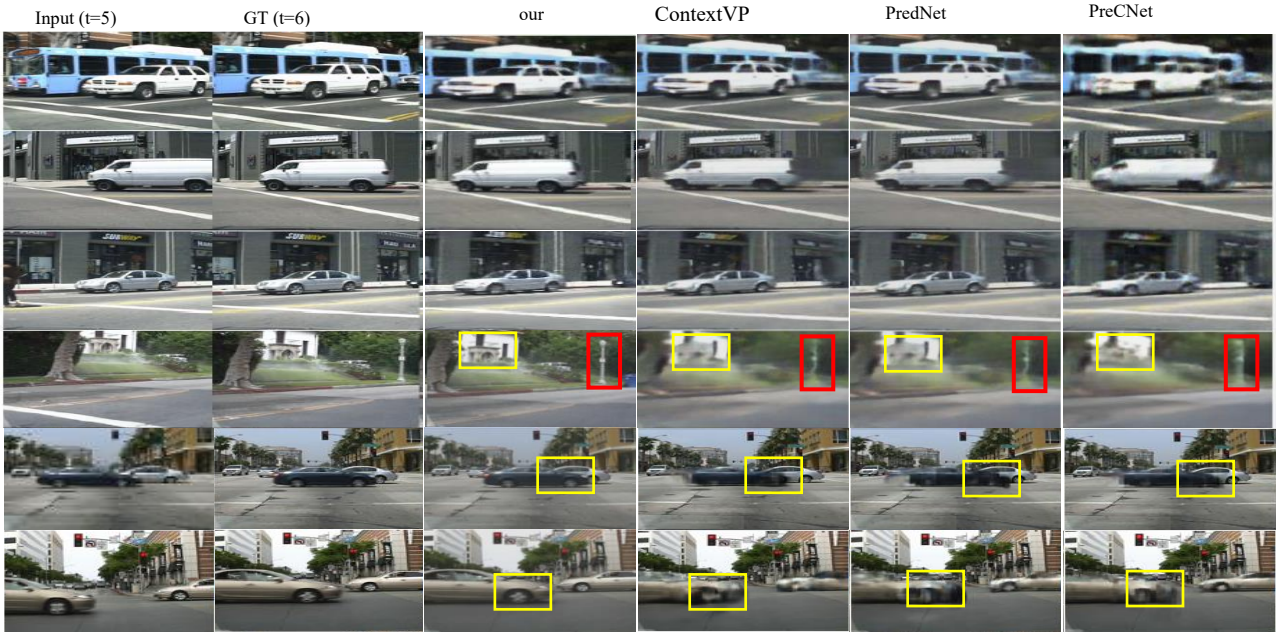

Figure 4. A qualitative comparison of PreCNet, PredNet, and Context VP models based on Caltech Pedestrian Dataset by rows: set07-v011, set10-v010, set10-v010, set06-v009, set06-v001, and set07-v011.

Table 4. The performance on the caltech pedestrian dataset after training on the KITTI dataset.

Method	MSE (10^{-4})	PSNR	SSIM	#Parameters
CtrlGen [13]	-------	20.88	0.766	-------
Copy the last frame	79.5	23.2	0.779	-------
VAE(3D-CONLSTM) [28]	51.4	----	0.864	12.9M
Mask ViT [12]	-------	26.2	0.407	228M
PredNet [21]	31.3	25.8	0.0.88	6.9M
DM-GAN [19]	24.1	--------	0.899	113M
PreCNet [30]	20.5	28.4	0.929	7.6M
ContextVP [2]	19.4	28.7	0.921	8.6M
Res GAN [22]	18.8	28.7	0.913	3.9M
RC-GAN [17]	16.1	29.2	0.919	------
FPNet-OF [27]	15.7	30.8	0.947	------
Our	4.8589	34.8673	0.9616	5.2 M

Temporal KL is applied in time space only at every training iteration. This helps to capture the proper spatial-temporal features based on the proposed 3D probability distribution. While the classical 2D-VAE models just applied the distribution probability on 2D context space. The 3D VAE produces the single frame prediction in 21 msec/step which can help to apply this model in a real-time system.

Figure 4 shows the qualitative results of different approaches based on the KITTI and Cityscapes datasets. The PredRNN [20], PreCNet [30], and ContextVP [2] models produced a good result but suffered from blurry prediction, especially at the motion object as shown in red and yellow areas in Figure 4. These models failed to

capture the proper temporal features. The basic goal of the 3D VAE model is to decrease the blurry prediction as small as possible with a minimum number of parameters. The results of 3D VAE are better than other methods because the 3D VAE model preserved the details and edges of the frame(s).

6.2. Multiple Frame Prediction Performance

The quantitative results of multi-frame prediction are evaluated based on cityscapes and KITTI datasets in the training phase and the Caltech pedestrian dataset in the testing phase. The prediction performance of the 3D VAE model is compared with other state-of-the-art next-frame prediction models, such as PredNet [20], dual motion GAN [18], PreCNet [30], retro-cycle GAN [17], and FPNet-OF [28]as shown in Table 5.

Table 5. The Quantitative analysis for the multi-frame prediction model.

Method		t=6	t=7	t=8	t=9	t=10	t=11
Dual-Motion GAN [18]	PSNR	-----	------	------	-----	------	------
	SSIM	0.90	0.89	0.88	0.87	0.86	------
PredNET [21]	PSNR	27.6	------	21.7	------	------	20.3
	SSIM	0.90	------		-------	-------	0.66
RC-GAN [22]	PSNR	29.2	------	25.9	-------	-------	22.3
	SSIM	0.91	-------	0.83	-------	-------	0.73
PreCNet [30]	PSNR	28.5	-------	23.4	-------	------	20.2
	SSIM	0.93	-------	0.82	-------	------	0.69
FPNet-OF [27]	PSNR	30.8	29.9	27.9	24.3	23.2	22.9
	SSIM	0.95	0.929	0.88	0.83	0.80	0.79
Our approach	PSNR	34.8	33.32	32.0	30.58	29.47	27.6
	SSIM	0.96	0.942	0.92	0.87	0.84	0.79

The input-output batch consists of eleven frames, the first five for input and the last six for ground truth. Each frame is resized to the 256×256 resolution and all pixels are normalized in the range of 0 and 1. First, the next frame is predicted based on a single frame prediction model, and then concatenate the result with the other four input frames to create a new input sequence and forecast the second next frame. This procedure of multi-frame prediction is repeated until the prediction frame contains a lot of blurry results.

From Table 5, at t=6, 7 and 8, the SSIM and PSNR have the highest value compared with the state-of-the-art models. But, at t=9 and 10, the dual motion GAN [37] recorded the highest value in SSIM, this is because of the low degradation in the metric based on adversarial training (the degradation loss in SSIM reaches 0.01). However, the degradation rate in the 3D VAE model increased for further frame prediction. It reaches to 0.051 in SSIM at t=11.

The predicted frames look good as shown in Figure 5, thus, the 3D VAE model outperforms all other models by a large margin in PSNR for the next frame prediction until t=11. The high value of SSIM is recorded for the sixth, seventh, and eighth future frames. This means that the 3D-VAE model obtains good results in the short-term forecasting horizon. Unfortunately, when updating the model to predict further and further frames, the prediction performance decreases drastically in SSIM. Therefore, we limit the number of outputs to 6 frames only.

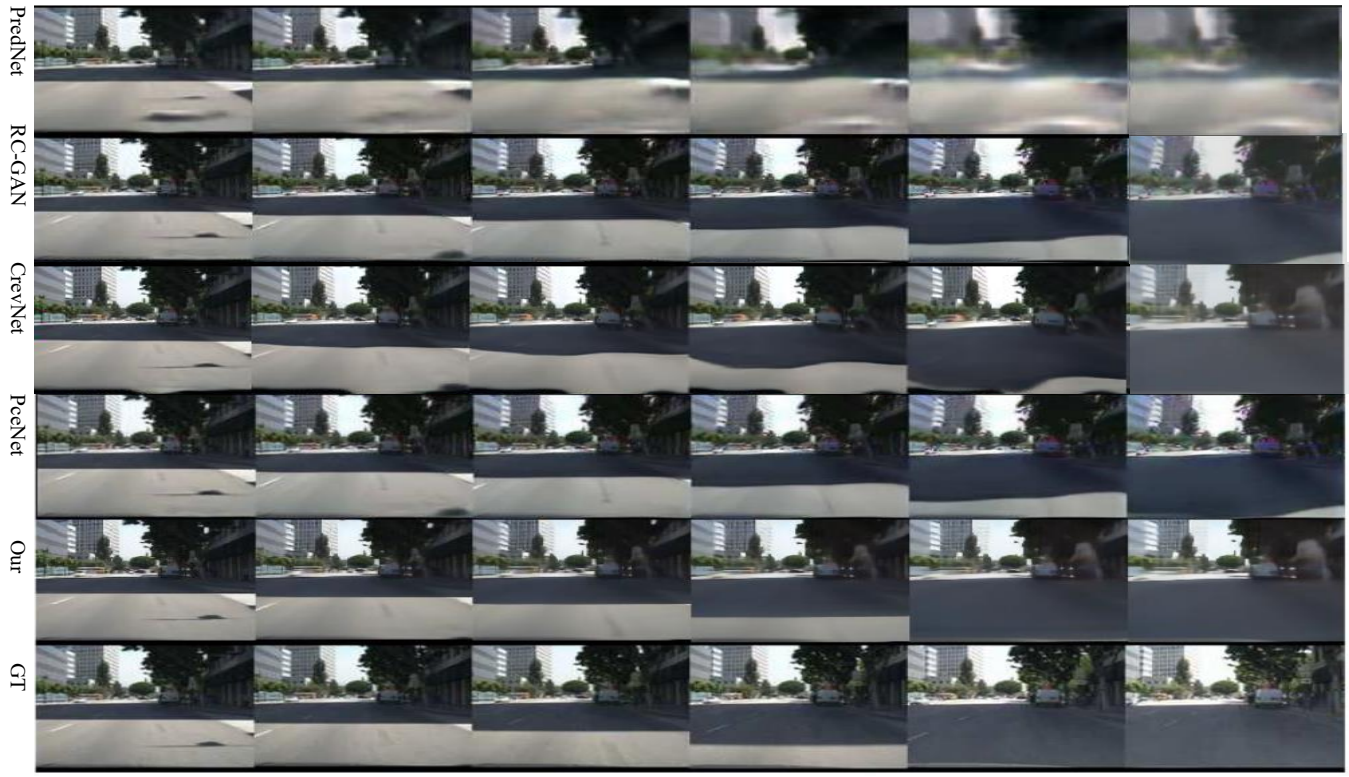

Figure 5. A qualitative evaluation of multi frame prediction algorithms that were selected. The inputs of PredNet and PreCNet are 10 frames, and RC-GAN used 4 input sequences. Our approach is applied 5 input frames. Location of the sequence in Caltech Pedestrian Dataset is set10-v009.

7. Conclusions

In this work, we introduced the end-to-end DL architecture of the 3D VAE model to predict the future video frame. The proposed model consists of a 3D Encoder, 3D Decoder parts, and 3D latent variable sampling and generates superior future frame prediction compared to other state-of-the-art architecture. The 3D VAE applies 3D CONV layers in Encoder and Decoder parts and 3D CONVLSTM to calculate the mean and variance value from latent space to create the 3D sampling. Experimentally, the effects of these two contributions are: First, the utilization of 3D CONVLSTM layers to calculate latent variables helps to create very well predictions beyond the time steps. The 3D KL_loss and Temporal KL_loss enables the model to produce better forecasting without an increase in model complexity by updating the weights of the whole model based on the normal distribution of the latent variable. Second, the 3D CNN layers extract the spatial-temporal features simultaneously which is very important to decrease the fuzz prediction results. The most important key in this study is that the prediction time based on single-frame prediction is more desirable for real-world applications. In future, we can apply the proposed model in GAN to build 3D VAE-GAN model based on the 3D KL_loss function.

Acknowledgement

The authors are thankful to anonymous reviewers and editors for their valuable comments.

References

[1] Asperti A. and Trentin M., "Balancing Reconstruction Error and Kullback-Leibler Divergence in Variational Autoencoders," *IEEE Access*, vol. 8, no. 1, pp. 199440-199448, 2020. DOI:10.1109/ACCESS.2020.3034828

[2] Byeon W., Wang Q., Srivastava R., and Koumoutsakos P., "ContextVP: Fully Context-Aware Video Prediction," *in Proceedings of the European Conference on Computer Vision*, Munich, pp. 753-769, 2018. https://doi.org/10.1007/978-3-030-01270-0_46

[3] Castrejon L., Ballas N., and Courville A., "Improved Conditional VRNNs for Video Prediction," *in Proceedings of the IEEE/International Conference on Computer Vision*, Seoul, pp. 7608-7617, 2019. https://doi.org/10.48550/arXiv.1904.12165

[4] Cheng Z., Sun H., Takeuchi M., and Katto J., "Deep Convolutional AutoEncoder-based Lossy Image Compression," *in Proceedings of the Picture Coding Symposium*, San Francisco, pp. 253-257, 2018. DOI:10.1109/PCS.2018.8456308

[5] Cordts M., Omran M., Ramos S., Rehfeld T., Enzweiler M., Benenson R., Franke U., Roth S., and Schiele B.,"The Cityscapes Dataset," *CVPR Work Future Datasets*, vol. 2, pp. 3213-3223, 2015. DOI:10.1109/CVPR.2016.350

[6] Courtney L. and Sreenivas R., "Comparison of Spatiotemporal Networks for Learning Video Related Tasks," *arXiv Preprint*, vol. arXiv:2009.07338v1, pp. 1-1, 2020. DOI:10.48550/arXiv.2009.07338

[7] Desai P., Sujatha C., Chakraborty S., Ansuman S., Bhandari S., and Kardiguddi S., "Next Frame Prediction Using ConvLSTM," *in Proceedings of the 1st International Conference on Artificial Intelligence, Computational Electronics and Communication System*, Manipal, pp. 28-30, 2021. DOI:10.1088/1742-6596/2161/1/012024

[8] Dollar P., Wojek C., Schiele B., and Perona P., "Pedestrian Detection: A Benchmark," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Miami, pp. 304-311, 2010. DOI:10.1109/cvpr.2009.5206631

[9] Gao Z., Tan C., Wu L., and Li S., "SimVP: Simpler Yet Better Video Prediction," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, New Orleans, pp. 3170-3180, 2022. DOI:10.1109/CVPR52688.2022.00317

[10] Geiger A., "Vision Meets Robotics: The KITTI Dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231-1237, 2013. doi: 10.1177/0278364913491297

[11] Goliński A., Pourreza R., Yang Y., Sautière G., and Cohen T., "Feedback Recurrent Autoencoder for Video Compression," *in Proceedings of the 15th Asian Conference on Computer Vision*, Kyoto, pp. 591-607, 2021. DOI:10.1007/978-3-030-69538-5_36

[12] Gupta A., Tian S., Zhang Y., Wu J., Martín R., and Fei L., "MaskViT: Masked Visual Pre-Training for Video Prediction," *in Proceedings of the 11th* International *Conference on Learning Representations*, Kigali, pp. 1-24, 2022. https://openreview.net/pdf?id=QAV2CcLEDh

[13] Hao Z., Huang X., and Belongie S., "Controllable Video Generation with Sparse Trajectories," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, pp. 7854-7863, 2018. DOI:10.1109/CVPR.2018.00819

[14] Hou X., Sun K., Shen L., and Qiu G., "Improving Variational Autoencoder with Deep Feature Consistent and Generative Adversarial Training," *Neurocomputing*, vol. 341, no. 14, pp. 183-194, 2019. https://doi.org/10.1016/j.neucom.2019.03.013

[15] Hu W., Li H. C., Pan L., Li W., Tao R., and Du Q., "Spatial-Spectral Feature Extraction via Deep ConvLSTM Neural Networks for Hyperspectral Image Classification," *IEEE Transactions on*

Geoscience and Remote Sensing, vol. 58, no. 6, pp. 4237-4250, 2020. DOI:10.1109/TGRS.2019.2961947

[16] Kapoor S., Sharma A., Verma A., Dhull V., and Goyal C., "A Comparative Study on Deep Learning and Machine Learning Models for Human Action Recognition in Aerial Videos," *The International Arab Journal of Information Technology*, vol. 20, no. 4, pp. 567-74, 2023. https://doi.org/10.34028/iajit/20/4/2

[17] Kwon Y. and Park M., "Predicting Future Frames Using Retrospective Cycle GAN," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, pp. 1811-1821, 2019. DOI:10.1109/CVPR.2019.00191

[18] Liang X., Lee L., Dai W., and Xing E., "Dual Motion GAN for Future-Flow Embedded Video Prediction," *in Proceedings of the IEEE International Conference on Computer Vision*, Venice, pp. 1744-1752, 2017. https://doi.org/10.1109/iccv.2017.194

[19] Liu B., Chen Y., Liu S., and Kim H., "Deep Learning in Latent Space for Video Prediction and Compression," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, pp. 701-710 2021. DOI:10.1109/CVPR46437.2021.00076

[20] Lotter W., Kreiman G., and Cox D., "Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning," *arXiv Preprint*, vol. arXiv:1605.08104v5, pp. 1-18, 2017. https://doi.org/10.48550/arXiv.1605.08104

[21] Lu W., Cui J., Chang Y., and Zhang L., "A Video Prediction Method Based on Optical Flow Estimation and Pixel Generation," *IEEE Access*, vol. 9, pp. 100395-100406, 2021. doi:10.1109/ACCESS.2021.3096788

[22] Lu Y., Mahesh Kumar K., Seyed Shahabeddin N., and Wang Y., "Future Frame Prediction Using Convolutional VRNN for Anomaly Detection," *in Proceedings of the 16th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Taipei, pp. 1-8, 2019. DOI:10.1109/AVSS.2019.8909850

[23] Michelucci U., "An Introduction to Autoencoders," *arXiv Preprint*, vol. arXiv:2201.03898v1, 2022. http://arxiv.org/abs/2201.03898

[24] Odaibo S., "Tutorial : Deriving the Standard Variational (VAE) Loss Function," *arXiv Preprint*, arXiv1907.08956, no. 2019, pp. 1-8, 2019. https://doi.org/10.48550/arXiv.1907.08956

[25] Oprea S., Martinez-Gonzalez P., Garcia-Garcia A., Castro-Vargas J., Orts-Escolano S., Garcia-Rodriguez J., and Argyros A., "A Review on Deep Learning Techniques for Video Prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 16, pp. 2806-2832, 2020. https://doi.org/10.1109/TPAMI.2020.3045007

[26] Pan J. Wang C., Jia X., Shao J., Sheng L., Yan J., and Wang X., "Video Generation from Single Semantic Label Map," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, pp. 3733-3742, 2019. DOI: 10.1109/CVPR.2019.00385

[27] Pratella D., Saadi S., Bannwarth S., Paquis-Fluckinger V., and Bottini S., "A Survey of Autoencoder Algorithms to Pave the Diagnosis of Rare Diseases," *International Journal of Molecular Sciences*, vol. 22, no. 19, pp. 1-14, 2021. doi:10.3390/ijms221910891

[28] Ranjan N., Bhandari S., Kim Y., and Kim H., "Video Frame Prediction by Joint Optimization of Direct Frame Synthesis and Optical-Flow Estimation," *Computers, Materials and Continua*, vol. 75, no. 2, pp. 2615-2639, 2023. DOI:10.32604/cmc.2023.026086

[29] Razali H. and Fernando B., "A Log-Likelihood Regularized KL Divergence for Video Prediction with a 3D Convolutional Variational Recurrent Network," *in Proceedings of the IEEE Winter Conference on Applications of Computer Vision* Workshops, Waikola, pp. 209-217, 2021. DOI:10.1109/WACVW52041.2021.00027

[30] Straka Z., Svoboda T., and Hoffmann M., "PreCNet: Next Frame Video Prediction Based on Predictive Coding," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 8, no. 57, pp. 1-23, 2023. http://arxiv.org/abs/2004.14878

[31] Suzuki M. and Matsuo Y., "A Survey of Multimodal Deep Generative Models," *Advanced Robotics*, vol. 36, no. 5-6, pp. 261-278, 2022. DOI:10.1080/01691864.2022.2035253

[32] Villegas R., Yang J., Hong S., Lin X., and Lee H., "Decomposing Motion and Content for Natural Video Sequence Prediction," *arXiv Preprint*, vol. arXiv1706.08033, pp. 1-22, 2018. http://arxiv.org/abs/1706.08033

[33] Wang Y., Jiang L., Yang M., Li L., Long M., and Fei-Fei L., "Eidetic 3d LSTM: A Model for Video Prediction and Beyond," *in Proceedings of the International Conference on Learning Representations*, New Orleans, pp. 1-14, 2019. https://openreview.net/pdf?id=B1lKS2AqtX

[34] Wei R. and Mahmood A., "Recent Advances in Variational Autoencoders with Representation Learning for Biomedical Informatics : A Survey," *IEEE Access*, vol. 9, pp. 4939-4956, 2021. DOI:10.1109/ACCESS.2020.3048309

[35] Wu B., Nair S., Martín-Martín R., Fei-Fei L., and Finn C., "Greedy Hierarchical Variational Autoencoders for Large-Scale Video Prediction," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, pp. 2318-2328, 2021. DOI:10.1109/CVPR46437.2021.00235

[36] Yang Y., Zheng K., Wu C., and Yang Y., "Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational Autoencoder And Deep Neural Network," *Sensors*, vol. 19, no. 11, 2019. DOI:10.3390/s19112528

[37] Ye X., Bilodeau G., and Montr P., "A Unified Model for Continuous Conditional Video Prediction," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vancouver, pp. 3603-3612, 2023. DOI:10.1109/CVPRW59228.2023.00368

Shefa Dawwd is currently a professor at the computer engineering department -university of Mosul. He received the B.Sc in Electronic and Communication Engineering, M.Sc and Ph.D. in Computer Engineering. He has authored more than 57 research papers and two book chapters. His research interest is in the Processing Acceleration of 1D, 2D, and 3D signals, Parallel Architecture, Real Time Applications, Deep Neural Networks, and Heterogeneous Computing.

Zahraa Al Mokhtar is currently working as a lecturer signals and system, University of Mosul, Mosul, Iraq. Her area of interest includes Image Processing, Processing of 1D, 2D, and 3D Signals Based on VHDL and Deep Learning Techniques.