IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 7, 2020, accepted March 30, 2020, date of publication April 10, 2020, date of current version April 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2987281

Deep Learning in Next-Frame Prediction: A Benchmark Review

YUFAN ZHOU, HAIWEI DONG, (Senior Member, IEEE), AND ABDULMOTALEB EL SADDIK, (Fellow, IEEE)

Multimedia Computing Research Laboratory, School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada

Corresponding author: Haiwei Dong (haiwei.dong@ieee.org)

ABSTRACT As an unsupervised representation problem in deep learning, next-frame prediction is a new, promising direction of research in computer vision, predicting possible future images by presenting historical image information. It provides extensive application value in robot decision making and autonomous driving. In this paper, we introduce recent state-of-the-art next-frame prediction networks and categorize them into two architectures: sequence-to-one architecture and sequence-to-sequence architecture. After comparing these approaches by analyzing the network architecture and loss function design, the pros and cons are analyzed. Based on the off-the-shelf data-sets and the corresponding evaluation metrics, the performance of the aforementioned approaches is quantitatively compared. The future promising research directions are pointed out at last.

INDEX TERMS Frame prediction architecture, loss function design, state-of-the-art evaluation.

I. INTRODUCTION

Next-frame prediction, that is, predicting what happens next in the form of an image or a few images, is an emerging field of computer vision and deep learning. This prediction is built on the understanding of the information in the historical images that have occurred so far. It refers to starting from continuous, unlabeled video frames and constructing a network that can accurately generate subsequent frames. The input of the network is the previous few frames, and prediction is/are the next frame(s). These predictions can be not only of human motion but also for any object motion and background in the images. 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 scenes in real life can be predicted since they satisfy physical laws (e.g., inertia), such as ball parabola prediction for a ping-pong robot [1]. The prediction of moving objects facilitates the advance decision of the machine. Similarly, images can also be predicted so that the machine can better understand the existing environment [2]. 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. Oh *et al.* [4] combined a predictive model with the deep Q-learning algorithm for better performance of an artificial intelligence (AI) agent in playing Atari games. Other approaches [5], [6] provided a visual predictive system for vehicles, which predicts a future position of pedestrians in the image to guide the vehicles to slow down or brake. Benefiting from next-frame prediction, Klein *et al.* [7] forecasted weather by predicting radar cloud images. In summary, next-frame prediction enables artificial intelligence to create a better understanding of the surrounding environments and provides a huge potential to deal with many different tasks based on predictive ability.

Since deep learning has shown its effectiveness in image processing [8], deep learning for next-frame prediction is very powerful compared with the traditional machine learning. Traditional machine learning methods often require the manual extraction of features and much preprocessing work. Time sequence predictions in machine learning use linear models such as ARIMA or exponential smoothing, which

The associate editor coordinating the review of this manuscript and approving it for publication was Mehedi Masud.

often fail to meet the needs of increasingly complex real-world environments. It is difficult to learn the features from images efficiently. A large number of methods have proven that deep learning is more suitable in the study of image representation learning [9]. Despite the great progress in deep-learning architecture, next-frame prediction remains a big challenge which can be summarized from two aspects: image deblurring and long-term prediction. In this paper, we define a prediction system as a long-term prediction if it can generate more than 20 future frames. The details will be discussed in the next sections. To conclude, the next-frame prediction is of great importance in the field of artificial intelligence by predicting future possibilities and making decisions in advance.

In this paper, we cover recent papers and discuss the ideas, contributions, and drawbacks of the previously proposed methods. Through the comparison of network structure, loss function design, and performance in experiments, the advantages and disadvantages of these methods are summarized, which inspires us to have a perspective towards the future research directions.

II. RELATED WORK

A. PREDICTIVE LEARNING

Predictive learning predicts future possibilities by understanding existing information. Generally, predictive learning is used to solve the sequence problem. There are several practical applications for using predictive learning. Recurrent networks are suitable for seeking patterns in sequence data, such as video representation, traffic flow, and weather prediction. Song *et al.* [22] proposed pyramid dilated bidirectional ConvLSTM to effectively detect significant regions in videos. Zhang *et al.* [23] predicted traffic flow by designing a spatiotemporal model. Shi *et al.* [24] predicted rainfall by the use of their proposed convolutional LSTM network (ConvLSTM), which combined the convolutional operation with a recurrent layer.

Additionally, as a way to create strong artificial intelligence, predictive learning has been applied in the fields of motion prediction, such as action prediction and trajectory prediction. Vu *et al.* [25] proposed a method to predict human actions from static scenes using the correlation information between actions and scenes. Ryoo *et al.* [26] implemented probabilistic action prediction and used the integration histogram of spatiotemporal features to model how the feature distribution changes over time. Lan *et al.* [27] developed a max-margin learning framework and proposed a new representation called ''human movemes'' for action prediction. Walker *et al.* [28] used the optical flow algorithm to mark a video and then trained an optical flow prediction model that can predict the motion of each pixel.

Furthermore, the performance and safety of self-driving cars can be improved since the behavior of vehicles and pedestrians can be estimated in advance by predictive learning systems. Walker *et al.* [29] tried to select the optimal target through a reward function to model the motion trajectory of a car. Deo *et al.* [30] presented a recurrent model based on LSTM [31] for the prediction of vehicle movement and its trajectory under the condition of freeway traffic.

B. DEEP LEARNING IN IMAGE GENERATION

Image generation is generating new images based on an existing dataset. It is generally divided into two categories. One category is to generate images according to attributes, which are generally text descriptions. The other category is from image to image: taking the historical images as the input of the network to generate image(s) for specific purposes, such as denoising [32], super-resolution [33] and image style transfer [34].

The most common network structures in the field of image generation are autoencoders and generative adversarial networks (GANs) [35]. Autoencoders are the most popular network architecture to generate images. An autoencoder is usually composed of two parts: an encoder and a decoder. The encoder encodes the data into a latent variable, and the decoder reconstructs the latent variable into the original data. There are several variants of autoencoders, including sparse autoencoders [36] and variable autoencoders (VAEs) [37]. As an example, Mansimov *et al.* [38] used a VAE to iteratively draw pictures based on words in an article based on the recurrent neural network DRAW [39]. The feature of the text description is used as the input of the network to generate the required image.

A generative adversarial network (GAN) is a commonly used training model in image generation. There are two components in a GAN: a discriminative model and a generative model. Images are generated from the generative model, while the discriminative model is trained to maximize the probability of applying the correct label to both examples and samples from the generative model. The discriminative model learns the boundary between classes, while the generative part models the distribution of individual classes. Using a GAN can make the generated picture clearer. Several approaches [33], [34] have been proposed to successfully generate sharper images with GANs. Furthermore, different types of GANs have been designed to generate images, such as ImprovedGAN [40] and InfoGAN [41]. There are also combinations of different GANs to perform image generation tasks. Zhang *et al.* [42] proposed StackGAN to iteratively generate high-quality images. There are two-stage generators in their approach. The first stage produces low-resolution images, and the second stage produces high-resolution images based on the results of the first stage. ProgressiveGAN [43] trained 4×4 pixel generators and discriminators first and then gradually added additional layers to double the output resolution to 1024×1024.

III. STATE-OF-THE-ART APPROACHES

Next-frame prediction can be taken as a spatiotemporal problem. Given a sequence of images in continuous time steps, predicting the next frame is performed by time sequence

TABLE 1. Comparison between different next-frame prediction approaches.

Architecture	Approach	Number of Input Frames	Number of Predicted frames	Loss Function	Model	Code
Sequence-to-one	Vukotic [10]	1	Up to 15	L_2 loss	CNN autoencoder	–
	Xue [11]	1	1	KL-divergence and L_1 loss	CNN autoencoder	https://github.com/tfxue/visual-dynamics
	Liu [12]	2	1,2,3	L_1 loss	Pyramid of CNN autoencoder	https://github.com/liuziwei7/voxel-flow
	Liang [6]	5,10	10,20	L_1 and Adversarial loss	GAN+CNN autoencoder+ConvLSTM	–
	Mathieu [13]	4,8	1,8	L_2, L_1, adversarial and gradient difference loss	Pyramid of CNN+GAN	https://github.com/coupriec/VideoPredictionICLR2016
	Villegas [14]	10	up to 128	L_2 and Adversarial loss	LSTM+CNN autoencoder+GAN	https://github.com/ZhongxiaYan/video_prediction
Sequence-to-sequence	Michalski [15]	2, 3, 5	1, 2, 20	L_2 loss	Pyramid of autoencoder and LSTM	–
	Srivastava [16]	10	10	Cross-entropy and L_2 loss	LSTM autoencoder	https://github.com/mansimov/unsupervised-videos
	Denton [17]	5,10	10,20	Cross-entropy, L_2 and Adversarial loss	CNN+GAN+LSTM	https://github.com/ap229997/DRNET
	Oliu [18]	10	10	L_1 loss	CNN+GRU autoencoder	https://github.com/moliusimon/frnn
	Oh [4]	4, 11	1	L_2 loss	CNN autoencoder+LSTM	https://github.com/junhyukoh/nips2015-\action-conditional-video-prediction
	Finn [19]	2, 10	up to 20	L_2 loss	ConvLSTM autoencoder	https://github.com/tensorflow/models/tree/master/research/video_prediction
	Lotter [5]	10	up to 5	L_1 loss	ConvLSTM+CNN	https://github.com/coxlab/prednet
	Wang [20]	3, 10	up to 20	L_1 and L_2 loss	ST-LSTM	https://github.com/ujjax/pred-rnn
	Wang [21]	3, 10	up to 20	L_1 and L_2 loss	Casual LSTM	https://github.com/Yunbo426/predrnn-pp

learning. Surrounding environmental information is learned from a sequence of images, and the regularity of pixel changes between images is retrieved. In addition, for a specific image, the relationship between pixels is a significant factor to be considered when performing next-frame prediction. The key feature can be extracted from the spatial structure of the image by the position, appearance, and shape of the object. We categorize the networks for next-frame prediction into two architectures: sequence-to-one architecture and sequence-to-sequence architecture. For the former architecture, the input of the deep learning model is a set of frames in time-step order between t and $t+k$. The prediction is the next frame. For the latter architecture, the input is temporal frames that are separately fed into the neural network. Specifically, the frame in time step t is input to the deep learning model, and the prediction is the next frame in time step $t+1$. This operation is continuously conducted until the deep learning model achieves the frame in the $(t+m)$th time step. Sequence-to-one architectures focus on the spatial structure from the set of input frames while sequence-to-sequence architectures focus on the factor of the temporal sequence.

Table 1 lists a collection of recent representative next-frame prediction approaches. These state-of-the-art approaches are compared in terms of their learning model structure, the number of inputs, the number of predicted frames, and loss functions. Additionally, the corresponding code URLs are included in Table 1. A recurrent neural network is rarely used in the sequence-to-one architecture, while it is widely used in the sequence-to-sequence architecture. As a classic neural network in image processing, autoencoders are widely used in both types of architectures for next-frame prediction. Usually, the encoder can extract the spatial features from the previous frames, and the decoder can regress pixels and reconstruct the next frames. The main advantage of autoencoder models is that they reduce the amount of input information by extracting the most representative information from the original image and put the reduced information into the neural network for learning. In addition, the structure of the autoencoder can adapt to a few input variables. For the loss function, L_1, L_2, and adversarial loss are used in both architectures. Among them, the most frequently used loss function is L_2. For the number of predicted frames, Villegas *et al.* [14] predicted the largest number of frames: up to 128 frames. In contrast, most state-of-the-art approaches can only predict less than 20 frames. In the following subsections, the two architectures are illustrated respectively.

A. SEQUENCE-TO-ONE ARCHITECTURE

As mentioned in the definition of a sequence-to-one architecture, most approaches [6], [12], [13], [15] concatenate the

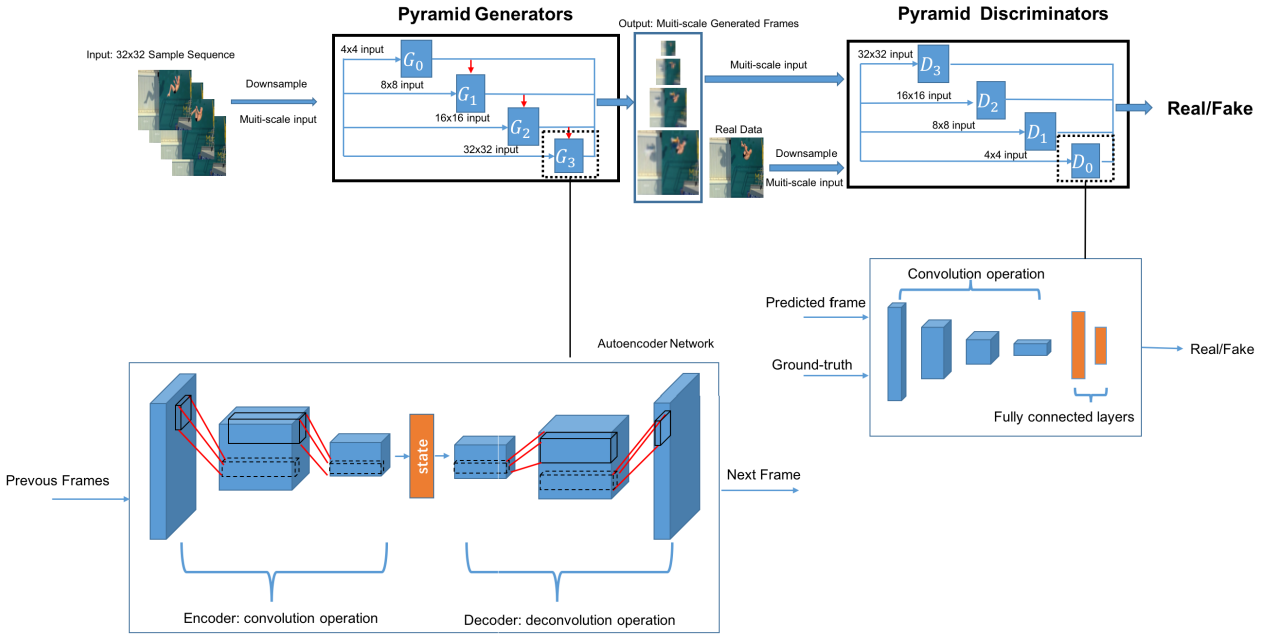


FIGURE 1. A general sequence-to-one architecture with a pyramid of autoencoder model and GAN. The generator is the autoencoder model where the autoencoder model consists of convolutional networks and the decoder consists of deconvolutional networks. The discriminator is a two-class classifier, which distinguishes the image generated by the generator from real data. The previous frames are resized into different sizes and fed into the overall network, while the output is the next frame with different scales.

previous frames on the channel dimension as a set of input images for networks. The frames are sorted in chronological order. Although the approaches from [10], [11] generated possible future frames from a single image, these two approaches are categorized as sequence-to-one architectures due to the following reasons. For the approach in [11], the authors converted the input frame into pyramid sizes. Therefore, the input to the image autoencoder is a multiscale set. Moreover, the authors set a motion autoencoder to extract the convolutional kernels from the image difference. The cross convolution operation from their approach combines the convolutional kernels with the feature maps from the image autoencoder. For the approach from [10], they considered the temporal factor as the input to the state layer in the autoencoder model. The encoder has two branches: one branch receives the input image and the other branch receives the time difference of the desired prediction. The decoder generates reliable frames based on the latent variable output from the encoder.

In image learning, the pyramid structure has been proven to be efficient for high-level feature extraction. As shown in Figure 1, feature pyramids are independently constructed from images of various scales. Feature pyramids are a semantically multiscale feature representation. Mathieu *et al.* [13] used multiscale networks through images at multiple scales to maintain a high resolution and reconstructed high-quality frames with. Liu *et al.* [12] proposed an end-to-end deep voxel flow network. They predicted the 3D voxel flow by a convolutional pyramid autoencoder. The voxel flow is added to a volume sampling layer to synthesize the desired frame. Nevertheless, featuring each scale of an image has an obvious limitation, i.e., slow processing speed. Lin *et al.* [44] proposed a feature pyramid network to improve the processing speed. Instead of extracting features from images of various scales, they extracted the various scales of features from a single image. The low-resolution and semantically strong features are combined with high-resolution and semantically weak features via a top-down pathway and lateral connections. They applied their approach to the task of object tracking. However, this kind of pyramid structure can be used to speed up the next-frame prediction model in the future research.

The combination of autoencoders and generative adversarial networks is a popular operation in the field of next-frame prediction, especially in the sequence-to-one architecture. Figure 1 is a representative example of a multiscale network with a GAN to generate the next predicted frame. In general, the input of the discriminative model is a real sample or a generated sample, whose purpose is to distinguish the output of the generative model from the real sample as much as possible. The two models confront each other and adjust the parameters continuously. The images generated by the approach in [13] are much sharper in their experiments with the help of a GAN. Since the set of generators in the approach [13] is a multiscale structure, the corresponding set of discriminators is also a multiscale structure. The loss calculated by multiple discriminators is accumulated and updated as the weights of the model. Hintz *et al.* [45] was inspired by Mathieu's method but replaced the generator with the reservoir computing network, which is a more complex RNN structure for dealing with high-dimensional sequence problems. The discriminator structure and training method remained

the same. In addition, the approaches of Liang *et al.* [6], Mathieu *et al.* [13], and Villegas *et al.* [14] used GANs to enhance the quality of the predicted frames. There are several kinds of GAN structures that can be used for next-frame prediction, such as WGAN [46], WAGAN-GP [47] and DCGAN [48]. Liang *et al.* [6] proposed dual WGANs to encourage background and motion prediction separately. Both decoders benefit from the learning of the other; a flow warping and a flow estimator are used to obtain the image from the predicted flow and vice versa, respectively. Therefore, they used two discriminators to distinguish real/fake frames and flows separately, ensuring that the pixel-level predictor produced sequences with proper motion and that the flow predictor was coherent at the pixel level.

Inspired by human pose estimation, Villegas *et al.* [14] used high-order structural information to assist in the next-frame prediction of human activities, which are key human body joints. The human skeleton structure is extracted from the input image with an hourglass network [49]. Villegas *et al.* [14] also used LSTM to predict the location of key points in the next frame. A recurrent neural network (RNN) plays a minor role in sequence-to-one architecture and assists prediction. The skeleton information in the next frame is fused into the encoder network in the form of a heat map. The experiments from the paper showed that this type of video generation based on high-order structural information can effectively reduce error propagation and accumulation. However, this method has certain limitations. The background information remains unchanged, and it can only model changes in human motion. For human activities, retaining static objects while predicting motion is a valuable direction.

By measuring the distance between the generated sample and the real sample, researchers usually use the L_1 or L_2 distance. Using only the L_1 or L_2 distance as the loss function will result in a blurry generated image. When predicting more frames, this problem is even more serious in a sequence-to-one architecture. To solve the problem of image blurriness caused by using the L_1 or L_2 loss function, Mathieu *et al.* [13] proposed an image gradient difference loss, which penalizes the gradient inconsistency between the predicted sample and the real sample by introducing the difference in the intensity of the neighboring image.

B. SEQUENCE-TO-SEQUENCE ARCHITECTURE

Another type of next-frame prediction architecture is the sequence-to-sequence architecture. It better reflects the character of temporal information. As shown in Figure 2, sequence-to-sequence architectures lead to different losses in time steps since they predict one frame in each time step. The best results are achieved by assigning different weights to each time point, which is the main difference in the setting of the loss function between sequence-to-sequence architectures and sequence-to-one architectures. Considering both the spatial and temporal features is one main feature of sequence-to-sequence architectures. Many researchers use the structure of recurrent neural networks to model the temporal sequence data and discover the relationship in a sequence.

Most sequence-to-sequence architectures use LSTM or ConvLSTM for future frame generation. The current approaches use multiple structures to consider both spatial and temporal features and combine the autoencoder model or GAN model with the RNN model. The approaches in [16], [18] first applied the encoder to the whole input sequence and then unroll the decoder for multiple predictions. In this sense, the model has fewer computational requirements. Michalski *et al.* [15] used a pyramid of gated autoencoders for prediction and employed recurrent connections to predict potentially any desired length of time. The higher layers in their network model the changes from the transformations extracted from the lower layers among the frames. Finn *et al.* [19] tried to decompose the motion and content: two key components that generate dynamics in videos. Their network is built upon an autoencoder convolutional neural network and ConvLSTM for pixel-level prediction. ConvLSTM is a classic representation of spatiotemporal predictive learning. Lotter *et al.* [5] used ConvLSTM for their prediction architecture based on the concept of predictive coding. In their approach, the image prediction error can be transmitted in the network to achieve a better way to learn the frame representation. Villegas *et al.* [50] also proposed a motion-content network (MCNet) for separating the background and human motion. The network has two encoder inputs: one encoder receives the image sequence difference as the motion input and uses LSTM to model the motion dynamics, and the other encoder receives the last frame of the static image. After combining outputs from LSTM and outputs from the static image encoder, the convolutional decoder takes the combination and outputs the predicted frames. Learning the temporal changes in objects' features is a new direction for predicting frames, but it has lead to a relatively small revolutionary change. Another innovative proposal is disentangled representation [51], which means that a change in a single underlying factor of variation should lead to a change in a single factor in the learned representation. A disentangled representation should separate the distinct, informative factors of variations in the data [52]. The application of disentangled representation in next-frame prediction is that applying a recurrent model to the time-varying components enables future-frame prediction. Denton *et al.* [17] broke down each feature into narrowly defined variables and encoded them as separate dimensions: pose and content. Pose is the sequence of frames. Content represents human actions. The combination of features from pose and content encoders was input to LSTM networks for applying next-frame prediction. In addition, motivated by DeepMind's use of the Atari game for reinforcement learning (RL) problems, Oh *et al.* [4] proposed two spatiotemporal prediction architectures based on a deep network containing action variables. Based on their understanding, future frames are related not only to past frames but also to the current operation or behavior.

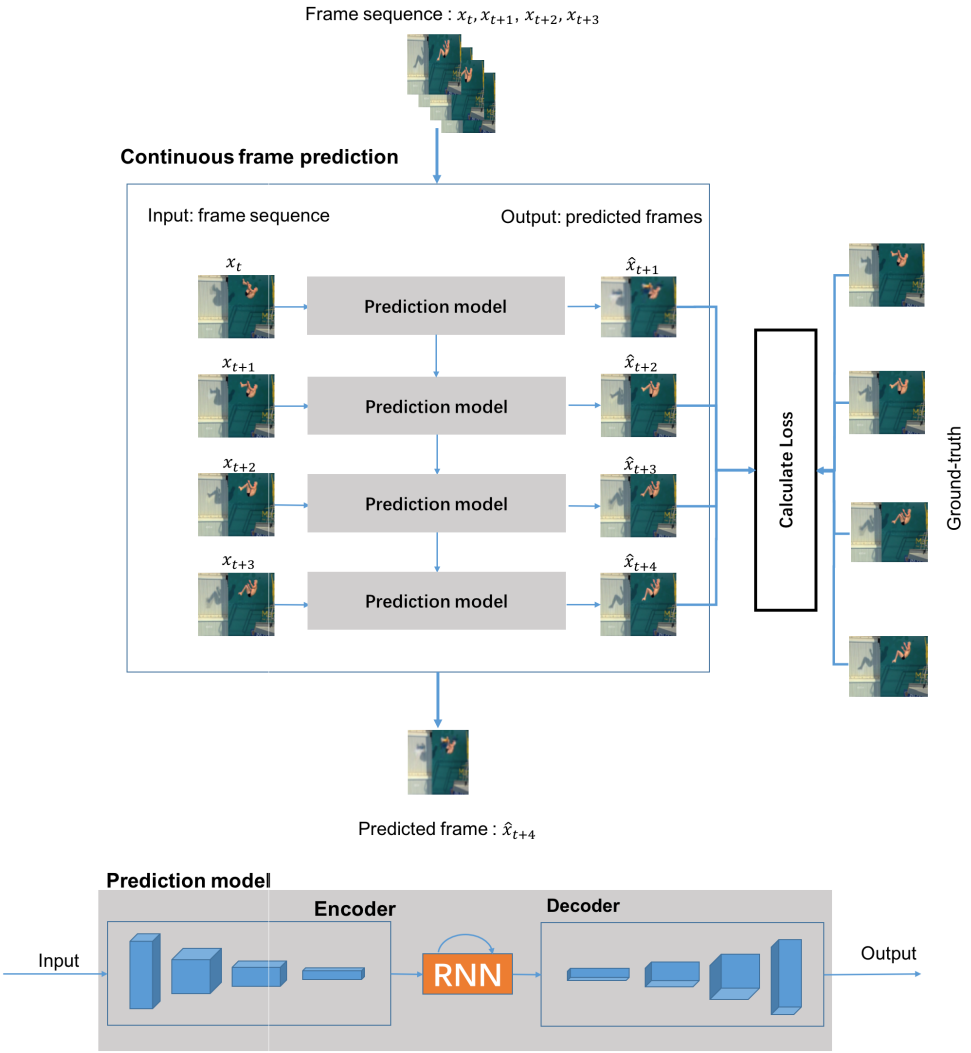


FIGURE 2. A general sequence-to-sequence architecture. The upper part is a representative expansion for sequence prediction. The lower part is the prediction model, which is composed of a recurrent neural network and an autoencoder. The frame sequence is fed into the RNN. The total loss is composed of the difference between the predicted images and the ground truth in each time step.

ConvLSTM is a classic representation of spatiotemporal predictive learning. Inspired by ConvLSTM, Wang *et al.* [20] designed a new strong RNN memory (called ST-LSTM) by adding the operation of spatiotemporal memory to the standard temporal cell. They made developments in the spatiotemporal structural information. A shared output gate is used in their ST-LSTM to fuse both memories seamlessly. In addition, they also proposed a new model structure: PredRNN. According to the authors, the traditional connection between multilayer RNN networks ignores the effect of the top-level cell at time t on the bottom-level cell at time $t+1$, and in their opinion, this effect is very significant. Therefore, they added top-level and bottom-level connections between the time steps t and $t+1$ in PredRNN. The combination of PredRNN and ST-LSTM can make the memory flow spread in both horizontal and vertical directions to make a high accuracy of long-term next-frame prediction. However, there is a disadvantage to performing long-term frame prediction in a sequence-to-sequence architecture. Since the predictions are made recursively, the small errors in the pixels are exponentially magnified when performing deeper future predictions. Based on the PredRNN, Wang *et al.* improved their network and proposed PredRNN++ [21] to make long-term next-frame predictions. In their model, a new spatiotemporal storage mechanism, called causal LSTM, was designed. The new LSTM attains more powerful modeling capabilities to achieve stronger spatial correlation and short-term dynamics. In addition, they also proposed a gradient highway unit, which provides a fast path for gradients from future predictions to long-interval past inputs to avoid the vanishing gradient problem.

IV. DATASETS AND EVALUATION METRICS

First of all, we define the symbols used in this section. The deep learning model generates the next frames $X_{n+1}, X_{n+2}, \ldots, X_{n+T}$, while the input is a set of continuous

video frames $X_1, X_2, \ldots, X_n$. T is the number of frames that need to be predicted. In this paper, we define Y and $\hat{Y}$ as the ground truth and the generated prediction, respectively. The prediction is $\hat{Y}$: $\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_T$.

Because it is a new field of research, we found that there are currently no datasets specifically designed for next-frame prediction, while researchers generally use motion video datasets or car-driving video datasets. We list eight commonly used datasets with their URLs, resolutions, motions, categories, videos and frame rates listed in Table 2. The above datasets are also used for action recognition, detection, and segmentation. The goal of next-frame prediction is to predict the changes in pixels between images. Due to smooth, continuous changes between frames from motion videos or car-driving videos, these datasets are the most appropriate for prediction. The Sports1m dataset has the most categories, while the Human3.6M dataset has the most videos. The frames per second (FPS) for all the datasets varies from 10 to 50. The image resolution of each dataset is different, from full-size images (2048×1024) to small-size images (640×360). Autonomous driving datasets, such as KITTI and CityScape, generally have a large image resolution. The categories of sports generally include walking, bowling, pushing up, jumping, bowling, diving, crawling, punching, swinging, hand waving, and hand clapping. Among the datasets, Sports1m, UCFsports and Penn Action contain sports scenes of athletes or people. Human3.6m, UCF101, THOMOS-15, and HMDB-51 datasets include not only sports scenes but also daily life scenes.

Since the final results are images, the commonly used methods for evaluating the quality of frames between the ground truth Y and the prediction $\hat{Y}$ are the mean square error (MSE), peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM) [53]. N is the number of pixels. The MSE measures the average of the squares of the errors or deviations. The MSE is calculated by:

$$MSE(Y, \hat{Y}) = \frac{1}{N} \sum_{t=1}^{N} (Y_i - \hat{Y}_i)^2 \tag{1}$$

The PSNR is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The PSNR is calculated by:

$$\text{PSNR}(Y, \hat{Y}) = 10 \log_{10} \frac{max_{\hat{Y}}^2}{\sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2} \tag{2}$$

where $max_{\hat{Y}}$ is the maximum possible value of the image intensities.

The SSIM measures the image similarity from luminance, contrast, and structure between two images. The calculation process is as follows:

$$SSIM(Y, \hat{Y}) = \frac{(2\mu_Y \mu_{\hat{Y}} + C_1) + (2\sigma_{Y\hat{Y}} + C_2)}{(\mu_Y^2 + \mu_{\hat{Y}^2} + C_1)(\sigma_Y^2 + \sigma_{\hat{Y}}^2 + C_2)} \tag{3}$$

where μ_Y and $\mu_{\hat{Y}}$ are the average of Y and $\hat{Y}$, respectively; σ_Y and $\sigma_{\hat{Y}}$ are the variance of Y and $\hat{Y}$, respectively; $\sigma_{Y\hat{Y}}$ is the covariance of Y and $\hat{Y}$. C_1 and C_2 are constants.

V. EXPERIMENTS AND DISCUSSION

A. EXPERIMENTS

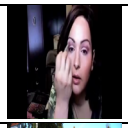
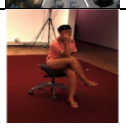
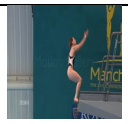
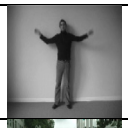
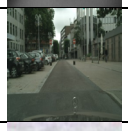
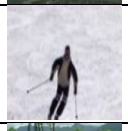
To conduct the experiments fairly, we set the time interval to 0.2 seconds between frames. There are two frequencies of the raw video data in our comparison experiment as 10 fps and 25 fps. We have modified the order of the input images. For a dataset of 25 fps, the inputs of the networks are the 1st frame, the 6th frame, the 11th frame, and the 16th frame, and the output is the prediction of the 21st frame. For the dataset of 10 fps, we use the following method: the inputs are the 1st frame, 3rd frame, 5th frame, and 7th frame, and the output is the prediction of the 9th frame. We construct the networks proposed by Denton, Liu, Mathieu, Srivasta, Oliu, Finn, Lotter, and Villegas on the UCFsports, KITTI, KTH, and UCF101 datasets. We use TensorFlow 1.12 and a GTX 1080 Ti GPU with 12 GB memory to train and test these networks and compare their prediction by using the MSE, PSNR, and SSIM. The results are shown in Table 3. The images are normalized between 0 and 1. The batch size is 32.

UCF101, KTH, and UCFsports represent the most challenging tasks in action prediction and classification. The KITTI dataset is a computer vision evaluation dataset from the autonomous driving platform AnnieWAY. The methods proposed by Villegas and Denton are required with human skeleton data for prediction, and KITTI is a vehicle driving dataset. Therefore, we did not train and test these models on the KITTI dataset. We have seen that all the methods can achieve a very high SSIM and PSNR values in predicting the frames in the action dataset experiment (KTH, UCF101, UCFsports) but low scores in the vehicle driving dataset experiment (KITTI). The reason for the low scores in the KITTI dataset is that objects in the image are complex: perhaps buildings, transportations, pedestrians and so on. We have implemented most of the temporal input architecture and three of the spatial input architecture. In the spatial input architecture, most of the methods use autoencoders to make predictions, but there are also special ones: Villegas considered the extra information of the human skeleton to predict the next frame, and other networks were not considered. Mathieu considers the pyramid network structure and can extract multiple features. Liu used a pyramid in the autoencoder network structure. In the sequence-to-sequence architecture, most of them use an RNN structure, and the difference is how to extract image features. Only Oh's method is not implemented because the prediction is the next frame of a video game based on the next action to play.

B. TECHNICAL ANALYSIS

From the results in Table 3, we can see that, in general, as the value of MSE decreases, both SSIM and PSNR increase.

TABLE 2. The commonly used datasets for the next-frame prediction.

Dataset	Image	Number of Categories	Videos	FPS	Resolution	Motion	URL
UCF101		101	13320	25	320x240	Diving, brushing teeth, dancing, baby crawling, bowling, punching, push-ups, climbing and so on	http://crcv.ucf.edu/data/UCF101.php
KITTI		–	–	10	1392x512	–	http://www.cvlibs.net/datasets/kitti/
Human3.6M		17	3.6 million	50	–	Conversations, eating, greeting, talking on the phone, posing, sitting, smoking, taking photos, waiting, walking in various atypical scenarios	http://vision.imar.ro/human3.6m/description.php
Sports1m		487	1.1 million	–	640x360	Boxing, bowling, ten-pin bowling, cycling, road bicycle racing, downhill mountain biking, jumping, skating and so on	https://cs.stanford.edu/people/karpathy/deepvideo/
Penn Action		15	2326	–	640x480	Baseball pitching, cleaning and jerking, pulling up, guitar strumming, baseball swinging, golf swinging, pushing up, forehand tennis, bench pressing, jumping, sitting, serving in tennis, bowling and squats	https://dreamdragon.github.io/PennAction/
UCFsports		10	150	10	720x480	Diving, golf swinging, kicking, lifting, riding a horse, running, skating boarding, swinging-bench, swinging-side and walking	http://crcv.ucf.edu/data/UCF_Sports_Action.php
KTH		6	600	25	160X120	Walking, jogging, running, boxing, hand waving and hand clapping	http://www.nada.kth.se/cvap/actions/
CityScape		–	–	17	2048x1024	–	https://www.cityscapes-dataset.com/
THUMOS-15		101	23100	–	–	Flipping, walking, riding, pulling, pushing, lifting, jumping up, bending, climbing up and so on.	THUMOS-15http://www.thumos.info/home.html
HMDB-51		51	6849	–	–	Hand clapping, climbing, diving, falling on the floor, flipping, jumping, pulling up, pushing up, running, sitting down and so on	http://serre-lab.clps.brown.edu/resource/hmdb-a-\large-human-motion-database/

Mathieu's achieved an outstanding result with an SSIM of 0.885 on the USFsports dataset, while Finn's method won first place on the KITTI, KTH, and UCF101 datasets. On the action datasets, some methods can achieve an SSIM of 80, but none of the results of the methods are ideal in the prediction of the autonomous driving dataset KITTI. The motion of each object in the autonomous driving scene is dynamic and complex. Predicting the next frame with the scene of vehicle driving is still a challenging problem.

Srivastava's method does not perform well since the fully connected network is not suitable for next-frame prediction. It cannot handle the diverse changes in the background, and many parameters require too much computation. GANs are helpful in the approaches proposed by Denton, Mathieu,

TABLE 3. The performance comparison of the state-of-the-art approaches in the next-frame prediction.

Approach	UCFsports			KITTI			KTH			UCF101		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
Denton [17]	0.0068	21.71	0.783	–	–	–	0.0038	24.43	0.816	0.0093	20.35	0.603
Liu [12]	0.0040	28.10	0.853	0.0112	20.82	0.659	0.0137	27.95	0.819	0.0094	23.27	0.685
Mathieu [13]	**0.0009**	**32.34**	**0.885**	0.0118	20.25	0.602	0.0034	27.62	0.845	0.0095	22.52	0.648
Srivastava [16]	0.1391	9.02	0.196	0.3234	5.34	0.102	0.0103	17.23	0.685	0.1791	8.35	0.175
Oliu [18]	0.0064	24.87	0.792	0.0195	17.31	0.501	0.0018	28.35	0.862	0.0091	22.86	0.739
Finn [19]	0.0033	28.43	0.855	**0.0076**	**22.25**	**0.679**	**0.0012**	**32.77**	**0.899**	**0.0036**	**28.04**	**0.846**
Lotter [5]	0.0323	23.13	0.718	0.0464	17.75	0.554	0.0137	27.95	0.819	0.0453	20.83	0.613
Villegas [14]	0.0094	23.32	0.735	–	–	–	0.0018	29.15	0.855	0.0104	22.46	0.652

Villegas, and Oliu. The desired predictions need to be guided under the discriminator. The reason for the low value is that Mathieu's method achieves a good prediction with fewer motions, whereas in the moving regions, it predicts frames with increased blurriness. As the results of Denton's approach show, simply using an autoencoder architecture to extract the features from the previous frames does not improve the prediction results. Finn's original approach is to predict future frames based on the constraints of action. Without the state of action, it achieves a low performance. Liu's method and Finn's method achieve a similar performance on the KITTI dataset. Since the data from the KITTI dataset are autonomous driving scenarios, the background is typically dynamic. The networks proposed by Lotter, Oliu, and Srivasta have limited ability to predict frames with dynamic backgrounds.

The temporal information of the objects also forms a part of the features of objects. Oliu and Lotter combined CNN and RNN models to extract the motion features recurrently. The main idea of these networks is combining CNN and RNN models that can recurrently extract motion features. The network always keeps the errors from each step to model the motion. After a few steps, the results are much better. The network of Oliu shares the information between the encoder and decoder to reduce the computational cost and avoids re-encoding the predictions when generating a sequence of frames.

C. DISCUSSION

Next-frame prediction is a new field of research for deep learning. In this paper, we have introduced the commonly used datasets, state-of-the-art and quantitative evaluations and conducted experiments on four datasets. Although these networks perform well, the existing methods still need to be improved. We describe some directions of improvement for next-frame prediction.

First, the combination of multiple network structures could improve next-frame prediction. Different networks address different problems. For example, CNNs are used to model spatial information, while RNNs are used to solve the time sequence problem. The combination of an RNN and a CNN can handle dynamic backgrounds and extract the necessary features from the previous frames. For next-frame prediction with multi-object motion (KITTI experiments), the current methods all achieved SSIMs of approximately 0.60, which indicates that there are large deviations in the generated images. In the image prediction for single object motion from the action datasets, the structure of the pyramid autoencoder, such as in Mathieu's method and Liu's method, is beneficial to maintaining the original feature information of the image. The structure of a CNN with LSTM can achieve a high SSIM and PSNR, such as with the methods proposed by Lotter, Oliu, and Finn. Finn used a multilayer ConvLSTM, while Lotter used a single layer of ConvLSTM. Although Oliu proposed a folded recurrent model, the essence of the network consists of an autoencoder and RNN structure. These results show that recurrent networks will play a positive role in next-frame prediction. All the researchers did not consider the time costs. Reducing the number of parameters as much as possible for fast prediction is of great value.

Second, the proper design of loss functions is another direction of improvement. The most commonly used loss functions are the mean squared error, GAN loss function, and image gradient differential loss function. Most networks use the per-pixel loss to measure the actual differences among pixels in the images, such as the mean squared error, mean average error and image gradient differential loss function. As shown in Table 3, using the per-pixel loss does not achieve a high-quality generated image. The reason may be that the constraint from per-pixel loss does not reflect the high-level features of the image. Different from the per-pixel loss, perceptual loss functions [54] are based on differences between high-level image feature representations since they make a large contribution in the field of image style transfer. The features extracted by convolutional neural networks can be used as parts of the loss function. By comparing the feature value from the image to be generated through the convolutional layers and the feature value from the target image through the convolutional layers, the generated image is more semantically similar to the target image. This is the main concept of the perceptual loss. As the purpose of next-frame prediction is to reconstruct future frames, the perceptual loss can play a significant role in the prediction. For human movement prediction, previous researchers did not consider the movement limitation of each body part. We can set up the loss function based on the movement of key points in the human body. During movement, the movement angles and distance between each point have a maximum and minimum

bounds. Based on the bounds, we can adjust the loss function to optimize the network directly.

Third, the current next-frame prediction approaches are pixel-level prediction. Each pixel value of both moving objects and static backgrounds is predicted. Such a prediction scheme needs a lot of computational resources. Differentiating the moving objects from the background can speed up the prediction. Additionally, combining the predicted motion with the original frames to perform next-frame prediction is another solution to improve the prediction efficiency. A good example is Villegas's work [14] which estimates the movement information of the human skeleton and transforms the skeletons into images to predict the next frame.

Fourth, a long-term prediction of future frames can be further improved. Most of the current next-frame prediction approaches can only predict short-term frames in the next one or two seconds. Regarding the predicted frames, to the best of our knowledge, the maximum number of the predicted frames is 128. In order to achieve longer-term prediction, more kinds of input information could be useful, such as depth image (compensating the 3D geometric information), infrared image (compensating the weak lighting condition), etc. By properly fusing the prediction results through optimal estimation (e.g., Kalman filter), a longer-term prediction may be able to achieved.

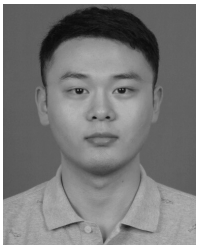
VI. CONCLUSION

Frame predictive learning is a powerful and useful way of understanding and modeling the dynamics of natural scenes. The long-term accurate prediction of the movement of an object, animal, or person is crucial to the future interactive human-machine interface, which can be widely applied in many areas, including simulating and predicting future road events, proactively cooperating with a human for robots, decision making and reasoning in understanding human's intention. As the SSIMs and PSNRs results of the state-of-the-art approaches are less than 0.9 and 30 respectively according to our experiments, we believe current research on next-frame prediction is still in the early stage. There is great potential for performance improvement in next-frame prediction.

REFERENCES

[1] H.-I. Lin and Y.-C. Huang, "Ball trajectory tracking and prediction for a ping-pong robot," in *Proc. 9th Int. Conf. Inf. Sci. Technol. (ICIST)*, Aug. 2019, pp. 222–227.
[2] Y. Miao, H. Dong, J. Al-Jaam, and A. El Saddik, "A deep learning system for recognizing facial expression in real-time," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 15, no. 2, pp. 3301–3320, 2019.
[3] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu, "Video pixel networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1771–1779.
[4] J. Oh, X. Guo, H. Lee, R. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in Atari games," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2863–2871.
[5] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–18.
[6] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion GAN for future-flow embedded video prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1762–1770.
[7] B. Klein, L. Wolf, and Y. Afek, "A dynamic convolutional layer for short range weather prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4840–4848.
[8] Y. Qiu, Y. Liu, J. Arteaga-Falconi, H. Dong, and A. El Saddik, "EVM-CNN: Real-time contactless heart rate estimation from facial video," *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1778–1787, Jul. 2019.
[9] Y. Jiang, H. Dong, and A. El Saddik, "Baidu Meizu deep learning competition: Arithmetic operation recognition using end-to-end learning OCR technologies," *IEEE Access*, vol. 6, pp. 60128–60136, 2018.
[10] V. Vukotic, S. L. Pintea, C. Raymond, G. Gravier, and J. V. Gemert, "One-step time-dependent future video frame prediction with a convolutional encoder-decoder neural network," in *Proc. Int. Conf. Image Anal. Process.*, 2017, pp. 140–151.
[11] T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 91–99.
[12] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4473–4481.
[13] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–14.
[14] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3560–3569.
[15] V. Michalski, R. Memisevic, and K. Konda, "Modeling deep temporal dependencies with recurrent grammar cells," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1925–1933.
[16] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using LSTMs," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1–10.
[17] E. Denton and V. Birodkar, "Unsupervised learning of disentangled representations from videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4414–4423.
[18] M. Oliu, J. Selva, and S. Escalera, "Folded recurrent neural networks for future video prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 716–731.
[19] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 64–72.
[20] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, "PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 879–888.
[21] Y. Wang, Z. Gao, M. Long, J. Wang, and P. S. Yu, "PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5123–5132.
[22] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, "Pyramid dilated deeper ConvLSTM for video salient object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 744–760.
[23] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proc. ACM Int. Conf. Adv. Geograph. Inf. Syst.*, 2016, pp. 1–4.
[24] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1, 2015, pp. 802–810.
[25] T. Vu, C. Olsson, I. Laptev, A. Oliva, and J. Sivic, "Predicting actions from static scenes," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 421–436.
[26] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1036–1043.
[27] T. Lan, T.-C. Chen, and S. Savarese, "A hierarchical representation for future action prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 689–704.
[28] J. Walker, A. Gupta, and M. Hebert, "Dense optical flow prediction from a static image," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2443–2451.
[29] J. Walker, A. Gupta, and M. Hebert, "Patch to the future: Unsupervised visual prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3302–3309.

[30] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1179–1184.
[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[32] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 341–349.
[33] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.
[34] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.
[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[36] A. Ng. (2010). *Sparse Autoencoder*. [Online]. Available: https://web.stanford.edu/class/cs294a/sparseAutoencoder.pdf
[37] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1–14.
[38] E. Mansimov, E. Parisotto, J. Ba, and R. Salakhutdinov, "Generating images from captions with attention," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–12.
[39] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "DRAW: A recurrent neural network for image generation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1462–1471.
[40] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.
[41] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.
[42] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5907–5915.
[43] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–26.
[44] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
[45] J. J. Hintz, "Generative adversarial reservoirs for natural video prediction," M.S. thesis, Univ. Texas Austin, Austin, TX, USA, 2016.
[46] M. Arjovsky, S. Chintala, and L. Bottou, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.
[47] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.
[48] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–16.
[49] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 483–499.
[50] R. Villegas, J. Yang, X. L. S. Hong, and H. Lee, "Decomposing motion and content for natural video sequence prediction," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–22.
[51] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1–37.
[52] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[53] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[54] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.



YUFAN ZHOU received the B.Eng. degree in railway traffic signaling and control from Southwest Jiaotong University, China, in 2017. He is currently pursuing the M.A.Sc. degree in electrical and computer engineering with the University of Ottawa. His research interests include artificial intelligence and multimedia.

HAIWEI DONG (Senior Member, IEEE) received the Dr.Eng. degree in computer science and systems engineering from Kobe University, Kobe, Japan, in 2008, and the M.Eng. degree in control theory and control engineering from Shanghai Jiao Tong University, Shanghai, China, in 2010. He was a Research Scientist with the University of Ottawa, Ottawa, ON, Canada; a Postdoctoral Fellow with New York University, New York, NY, USA; a Research Associate with the University of Toronto, Toronto; and a Research Fellow (PD) with the Japan Society for the Promotion of Science, Tokyo, Japan. He is currently a Principal Engineer with the Noah's Ark Lab of Huawei Technologies, Toronto, ON, Canada. His research interests include artificial intelligence, robotics, and multimedia.

ABDULMOTALEB EL SADDIK (Fellow, IEEE) is a Distinguished University Professor and a University Research Chair with the School of Electrical Engineering and Computer Science, University of Ottawa. He has supervised more than 120 researchers. He has coauthored ten books and more than 550 publications and chaired more than 50 conferences and workshops. His research focus is on the establishment of digital twins to facilitate the well-being of citizens using AI, the IoT, AR/VR, and 5G to allow people to interact in real time with one another as well as with their smart digital representations. He has received research grants and contracts totaling more than $20 M.

He is an ACM Distinguished Scientist and a Fellow of the Engineering Institute of Canada and the Canadian Academy of Engineers. He has received several international awards, such as the IEEE I&M Technical Achievement Award, the IEEE Canada C.C. Gotlieb (Computer) Medal, and the A.G.L. McNaughton Gold Medal for important contributions to the field of computer engineering and science.