ContextVP: Fully Context-Aware Video Prediction

Wonmin Byeon[1,2,3,4(✉)], Qin Wang[2], Rupesh Kumar Srivastava[4], and Petros Koumoutsakos[2]

[1] NVIDIA, Santa Clara, CA, USA
wbyeon@nvidia.com
[2] ETH Zurich, Zurich, Switzerland
[3] The Swiss AI Lab IDSIA, Manno, Switzerland
[4] NNAISENSE, Lugano, Switzerland

Abstract. Video prediction models based on convolutional networks, recurrent networks, and their combinations often result in blurry predictions. We identify an important contributing factor for imprecise predictions that has not been studied adequately in the literature: blind spots, i.e., lack of access to all relevant past information for accurately predicting the future. To address this issue, we introduce a fully context-aware architecture that captures the entire available past context for each pixel using Parallel Multi-Dimensional LSTM units and aggregates it using blending units. Our model outperforms a strong baseline network of 20 recurrent convolutional layers and yields state-of-the-art performance for next step prediction on three challenging real-world video datasets: Human 3.6M, Caltech Pedestrian, and UCF-101. Moreover, it does so with fewer parameters than several recently proposed models, and does not rely on deep convolutional networks, multi-scale architectures, separation of background and foreground modeling, motion flow learning, or adversarial training. These results highlight that full awareness of past context is of crucial importance for video prediction.

1 Introduction

Unsupervised learning from unlabeled videos has recently emerged as an important direction of research. In the most common setting, a model is trained to predict future frames conditioned on the past and learns a representation that captures information about the appearance and the motion of objects in a video without external supervision. This opens up several possibilities: the model can be used as a prior for video generation, it can be utilized for model-based reinforcement learning [32], or the learned representations can be transferred to other video analysis tasks such as action recognition [30]. However, learning such predictive models for natural videos is a rather challenging problem due to the

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01270-0_46) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11220, pp. 781–797, 2018.
https://doi.org/10.1007/978-3-030-01270-0_46

diversity of objects and backgrounds, various resolutions, object occlusion, camera movement, dynamic scene and light changes between frames. As a result, current video prediction models based on convolutional networks, recurrent networks, and their combinations often result in imprecise (blurry) predictions. Even very large, powerful models trained on large amounts of data can suffer from fundamental limitations that lead to blurry predictions. The structure of certain models may be inappropriate for the task, resulting in training difficulties and poor generalization. Some researchers have proposed to incorporate motion priors and background/foreground separation into model architectures to counter this issue.

Blurry predictions are fundamentally a manifestation of model uncertainty, which increases if the model fails to sufficiently capture relevant past information. Unfortunately, this source of uncertainty has not received sufficient attention in the literature. Most current models are not designed to ensure that they can properly capture all possibly relevant past context. This paper attempts to address this gap.

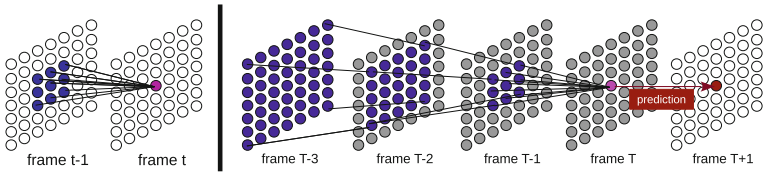

Fig. 1. (left) The Convolutional LSTM (ConvLSTM) context dependency between two successive frames. (right) The context dependency flow in ConvLSTM over time for frame $t = T$. Blind areas shown in gray cannot be used to predict the pixel value at time $T + 1$. Closer time frames have larger blind areas.

Our contributions are as follows:

- We highlight a **blind spot problem** in common video prediction models, showing that they do not systematically take the entire spatio-temporal context from past frames into account (see Fig. 1-right) and have to rely on increasing depth to do so. This increases uncertainty about the future which can not be remedied using special loss functions or motion priors.
- We contribute a **simple baseline model that outperforms rather complex models from recent literature**. Due to increased depth, this baseline model has an increased ability to capture relevant context.
- We propose a **new architecture for video prediction** that systematically and efficiently aggregates contextual information for each pixel in all possible directions (left, right, top, bottom, and time directions) at each processing layer (see Fig. 2) instead of stacking layers to cover the available context. We additionally propose *weighted context-blending blocks* and *regularization via directional weight sharing* for the proposed architecture. We obtain performance improvements over our strong baseline as well as state-of-the-art models while using fewer parameters and simple loss functions.

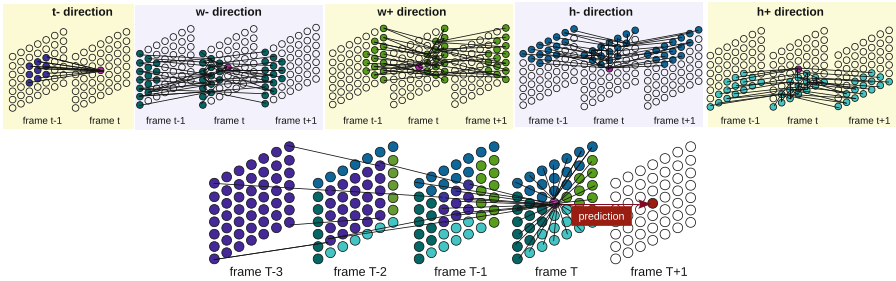

Fig. 2. (top) Context dependency between two frames when using Parallel MD-LSTM (PMD) units for five directions: $t-$, $w-$, $w+$, $h-$, and $h+$, where h, w, and t indicate the current position for height, width, and time dimensions. (bottom) The combined context dependency flow for frame $t = T$ in the proposed architecture. All available context from past frames is covered in a single layer regardless of the input size.

We demonstrate improvements in a variety of challenging video prediction scenarios: car driving, human motion, and diverse human actions in YouTube videos. Quantitative improvements on metrics are accompanied by results of high visual quality showing sharper future predictions with reduced blur or other motion artifacts. Since the proposed models do not require separation of content and motion or novel loss functions to reach the state of the art, we find that full context awareness is the crucial ingredient for high quality video prediction.

2 Related Work

Current approaches for video analysis exploit different amounts of spatio-temporal information in different ways depending on model architecture. One common strategy is to use models based on *3D Convolutional Neural Networks (CNNs)* that use convolutions across temporal and spatial dimensions to model all local correlations [28,33] for supervised learning. Similar architectures have been used for video prediction to directly generate the RGB values of pixels in future frames [22,24,26,35]. Kalchbrenner et al. [16] discussed that a general probabilistic model of videos should take into account the entire history (all context in past frames and generated pixels of present frame) for generating each new pixel. However, their proposed Video Pixel Networks (VPNs) still use encoders based on stacks of convolution layers. An inherent limitation of these models is that convolutions take only short-range dependencies into account due to the limited size of the kernels. These architectures need a larger stack of convolutional layers to use a wide context for reducing uncertainty. This increases the model capacity even though it may not be needed.

Recurrent neural networks are often used to address the issue of limited context. Srivastava et al. [30] proposed Long Short-Term Memory (LSTM) [13] based encoder-decoder models for the task of video prediction, but the canonical LSTM architecture used by them did not take the spatial structure of video

data into account. This motivated the use of *Convolutional LSTM (ConvLSTM)* based models which replace the internal transformations of an LSTM cell with convolutions. Xingjian et al. [38] proposed this design for precipitation nowcasting; the motivation being that the convolution operation would model spatial dependencies, while LSTM connectivity would offer increased temporal context. The same modification of LSTM was simultaneously proposed by Stollenga et al. [31] for volumetric image segmentation under the name PyraMiD-LSTM, due to its close relationship with the Multi-Dimensional LSTM (MD-LSTM) [12].

Recently, ConvLSTM has become a popular building block for video prediction models. Finn et al. [6] used it to design a model that was trained to predict pixel motions instead of values. Lotter et al. [21] developed the Deep Predictive Coding Network (PredNet) architecture inspired by predictive coding, which improves its own predictions for future frames by incorporating previous prediction errors. It is also used in the MCNet [34] which learns to model the scene content and motion separately, and in the Dual Motion GAN [19] which learns to produce consistent pixel and flow predictions simultaneously. Wang et. al. [36] have recently proposed the modification of stacked ConvLSTM networks for video prediction by sharing the hidden state among the layers in the stack.

For videos with mostly static backgrounds, it is helpful to explicitly model moving foreground objects separately from the background [6,28,35]. Another active line of investigation is the development of architectures that only learn to estimate optical flow and use it to generate future frames instead of generating the pixels directly [20,25].

Deterministic models trained with typical loss functions can result in imprecise predictions simply because the future is ambiguous given the past. For example, if there are multiple possible future frames, models trained to minimize the L2 loss will generate their mean frame. One approach for obtaining precise, natural-looking frame predictions in such cases is the use of adversarial training [22,35] based on Generative Adversarial Networks [9]. Another is to use probabilistic models for modeling the distribution over future frames, from which consistent samples can be obtained without averaging of modes [16,39].

3 Missing Contexts in Other Network Architectures

As mentioned earlier, blurry predictions can result from a video prediction model if it does not adequately capture all relevant information in the past video frames which can be used to reduce uncertainty. Figure 1 shows the recurrent connections of a pixel at time t with a 3×3 convolution between two frames (left) and the information flow of a ConvLSTM predicting the pixel at time $T + 1$ (right). The covering context grows progressively over time (depth), but there are also blind spots which cannot be used for prediction. In fact, as can be seen in Fig. 1 (right, marked in gray color), frames in the recent past have larger blind areas. Due to this structural issue, the network is unable to capture the entire available context and is likely to miss important spatio-temporal dependencies leading to increased ambiguity in the predictions. The prediction will eventually fail when

the object appearance or motion in videos changes dramatically within a few frames.

One possible way to address limited context, widely used in CNNs for image analysis, is to expand context by stacking multiple layers (sometimes with dilated convolutions [40]). However, stacking layers still limits the available context to a maximum as dictated by the network architecture, and the number of additional parameters required to gain sufficient context can be very large for high resolution videos. Another technique that can help is using a multi-scale architecture, but fixed scale factors may not generalize to all possible objects, their positions and motions.

4 Method

We introduce the Fully Context-aware Video Prediction model (ContextVP)—an architecture that avoids blind spots by covering all the available context by design. Its advantages are:

- Since each processing layer covers the entire context, increasing depth is only used as necessary to add computation power, not more context. A priori specification of scale factors is also not required.
- Compared to models that utilize increased depth to cover larger context such as our baseline 20-layer models, more computations can be parallelized.
- Compared to state-of-the-art models from recent literature, it results in improved performance without the use of separation of motion and content, learning optical flow or adversarial training (although combinations with these strategies may further improve results).

Let $x_1^T = \{x_1, ..., x_T\}$ be a given input sequence of length T. $x_t \in \mathbb{R}^{H \times W \times C}$ is the t-th frame, where $t \in \{1, ..., T\}$, H is the height, W the width, and C the number of channels. For simplicity, assume $C = 1$, x_1^T is then a cuboid of pixels bounded by six planes. The task is to predict p future frame(s) in the sequence, $x_{t+1}^{t+p} = \{x_{t+1}, ..., x_{t+p}\}$ (next-frame prediction if $p = 1$). Therefore, our goal is to integrate information from the entire cuboid x_1^T into a representation at the plane where $t = T$, which can be used for predicting x_{t+1}^{t+p}. This is achieved in the proposed model by using fully context-aware layers, each consisting of two blocks. The first block is composed of *Parallel MD-LSTM units* that sequentially aggregate information from different directions. The second block is the *Context Blending Block* that combines the output of PMD units for all directions. The context covered using PMD units for each direction (top) and the combined context from past frames (down) are visualized in Fig. 2. The schematic in Fig. 4 shows the overall architecture of our best model.

4.1 Parallel MD-LSTM Unit

Multidimensional LSTM (MD-LSTM) [12] networks, a specialization of DAG-RNNs [2], have been applied to various problems where the input is two-dimensional such as handwriting recognition [11], 2D image classification [4] and

segmentation [3]. They consist of two MD-LSTM blocks per dimension to combine context from all possible directions. In principle, MD-LSTM networks can be applied to any high-dimensional domain problem (including video prediction) to model all available dependencies in the data compactly. However, the fully sequential nature of the model makes it unsuitable for parallelization and thus impractical for higher dimensional data. The *PyraMiD-LSTM* [31] addressed this issue by re-arranging the recurrent connection topology of each MD-LSTM block from cuboid to pyramidal (for 3D data). It could be implemented efficiently by utilizing the convolution operation. So far, the idea of using LSTM to aggregate information from all directions was only explored in a limited setting (2D/3D image segmentation).

We refer to the parallel computing units used in the PyraMiD-LSTM architecture simply as Parallel Multi-Dimensional (PMD) units since they model contextual dependencies in a way that is amenable to parallelization. They are mathematically similar to ConvLSTM units but our terminology highlights that it is **not** necessary to limit convolutional operations to spatial dimensions and LSTM connectivity to the temporal dimension as is conventional. As can be seen in Fig. 3, PMD units can be used to aggregate context along any of the six directions available in a cuboid. Three directions are shown: $t-$, $w+$, and $h+$. At each plane, the local computation for each pixel is independent of other pixels in the same plane, so all pixels are processed as parallel using the convolution operation. The computational dependencies across planes are modeled using the LSTM operation. Computations for each PMD unit are explained mathematically below.

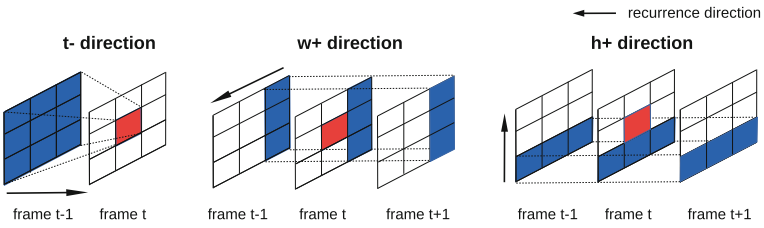

Fig. 3. Illustration of one computation step of PMD units for $t-$, $w+$, and $h+$ recurrence directions. Each unit computes the activation at the current position (red) using context from a fixed receptive field (here 3×3) of the previous frame along its recurrence direction (blue). This computation is efficiently implemented using convolutions.

For any sequence of K two dimensional planes $x_1^K = \{x_1, ..., x_K\}$, the PMD unit computes the current cell and hidden state c_k, s_k using input, forget, output gates i_k, f_k, o_k, and the transformed cell $\tilde{c}_k$ given the cell and hidden state from

the previous plane, c_{k-1}, s_{k-1}.

$$\begin{aligned} i_k &= \sigma(W_i * x_k + H_i * s_{k-1} + b_i), \\ f_k &= \sigma(W_f * x_k + H_f * s_{k-1} + b_f), \\ o_k &= \sigma(W_o * x_k + H_o * s_{k-1} + b_o) \\ \tilde{c}_k &= \tanh(W_{\tilde{c}} * x_k + H_{\tilde{c}} * s_{k-1} + b_{\tilde{c}}), \\ c_k &= f_k \odot c_{k-1} + i_k \odot \tilde{c}_k, \\ s_k &= o_k \odot \tanh(c_k). \end{aligned} \tag{1}$$

Here $(*)$ is the convolution operation, and $(\odot)$ the element-wise multiplication. W and H are the weights for input-state and state-state. The size of weight matrices are dependent only on the kernel size and number of units. If the kernel size is larger, more local context is taken into account.

As shown in Sect. 3, using a ConvLSTM would be equivalent to running a PMD unit along the time dimension from $k = 1$ to $k = T$, which would only integrate information from a pyramid shaped region of the cuboid and ignore several blind areas. For this reason, it is necessary to use four additional PMD units, for which the conditioning directions are aligned with the spatial dimensions, as shown in Fig. 2 (top). We define the resulting set of five outputs at frame T as s^d where $d \in D = \{h-, h+, w-, w+, t-\}$ denotes the recurrence direction. Together this set constitutes a representation of the cuboid of interest x_1^T. Outputs at other frames in x_1^{T-1} are ignored.

4.2 Context Blending Block

This block captures the entire available context by combining the output of PMD units from all directions at frame T. This results in the critical difference from the traditional ConvLSTM: the context directions are aligned not only with the time dimension but also with the spatial dimensions. We consider two ways to combine the information from different directions.

Uniform blending (U-blending): this strategy was used in the traditional MD-LSTM [3,10] and PyraMiD LSTM [31]. It simply sums the output of all directions along the channel dimension and then applies a non-linear transformation on the result:

$$m = f((\sum_{d \in D} s^d) \cdot W + b), \tag{2}$$

where $W \in \mathbb{R}^{N1 \times N2}$ and $b \in \mathbb{R}^{N2}$ are a weight matrix and a bias. $N1$ is the number of PMD units, and $N2$ is the number of (blending) blocks. f is an activation function.

Weighted blending (W-blending): the summation of PMD unit outputs in U-blending assumes that the information from each direction is equally important for each pixel. We propose W-blending to remove this assumption and learn the relative importance of each direction during training with the addition of a

small number of additional weights compared to the overall model size. The block concatenates s from all directions:

$$S = \left[s^{t-}\ s^{h-}\ s^{h+}\ s^{w-}\ s^{w+}\right]^T \tag{3}$$

The vector S is then weighted as follows:

$$m = f(S \cdot W + b), \tag{4}$$

where $W \in \mathbb{R}^{(5\times N1)\times N2}$ (5 is the number of directions). Equations 2 and 4 are implemented using 1×1 convolutions. We found that W-blending is crucial for achieving high performance for the task of video prediction (see Table 1).

4.3 Directional Weight-Sharing (DWS)

Visual data tend to have structurally similar local patterns along opposite directions. This is the reason why horizontal flipping is a commonly used data augmentation technique in computer vision. We propose the use of a similarly inspired weight-sharing technique for regularizing the proposed networks. The weights and biases of the PMD units in opposite directions are shared i.e. weights for $h-$ and $h+$ are shared, as are $w-$ and $w+$. This strategy has several benefits in practice: (1) it lowers the number of parameters to be learned, (2) it incorporates knowledge about structural similarity into the model, and (3) it improves generalization.

4.4 Training

$\hat{x} = g(m)$ is an output of the top-most (output) layer, where g is an output activation function. The model minimizes the loss between the predicted pixels and the target pixels. $\mathcal{L}_p$ loss and the Image Gradient Difference Loss (GDL) [22] are combined. By keeping the loss function simple, the results reflect the impact of having access to all available context. Let y and $\hat{x}$ be the target and the predicted frame. The objective function is defined as follows:

$$\begin{aligned}
\mathcal{L}(y,\hat{x}) &= \lambda_p \mathcal{L}_p(y,\hat{x}) + \lambda_{gdl}\mathcal{L}_{gdl}(y,\hat{x}) \\
\mathcal{L}_p(y,\hat{x}) &= ||y-\hat{x}||_p \\
\mathcal{L}_{gdl}(y,\hat{x}) &= \sum_{i,j} |y_{i,j}-y_{i-1,j}| - |\hat{x}_{i,j}-\hat{x}_{i-1,j}| + |y_{i,j-1}-y_{i,j}| - |\hat{x}_{i,j-1}-\hat{x}_{i,j}|,
\end{aligned} \tag{5}$$

where $|.|$ is the absolute value function, $\hat{x}_{i,j}$ and $y_{i,j}$ are the pixel elements from the frame $\hat{x}$ and y, respectively. λ_p and λ_{gdl} are the weights for each loss. In our experiments, λ_{gdl} is set to 1 when $p = 1$ and 0 when $p = 2$. λ_p is always set to 1.

We use ADAM [18] as the optimizer with an initial learning rate of $1e-3$. The learning rate is decayed every 5 epochs with the decay rate 0.99. Weights are initialized using the Xavier's normalized initializer [8] and the states of the LSTMs are initialized to zero (no prior).

5 Experiments

We evaluate the proposed approach on three real-world scenarios with distinct characteristics: human motion prediction (Human 3.6M dataset [14]), car-mounted camera video prediction (train: KITTI dataset [7], test: CalTech Pedestrian dataset [5], and human activity prediction (UCF-101 dataset [29]). All input pixel values are normalized to the range $[0, 1]$. For human motion and car-mounted videos, the models are trained to use ten frames as input for predicting the next frame. For the UCF-101 dataset, the input consists of four frames for fair comparison to past work. Quantitative evaluation on the test sets is performed based on mean Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM) [37][1]. These commonly used numerical measures are known to be not fully representative of human vision. Therefore, we highly recommend looking at the visual results in Figs. 5 and 7.

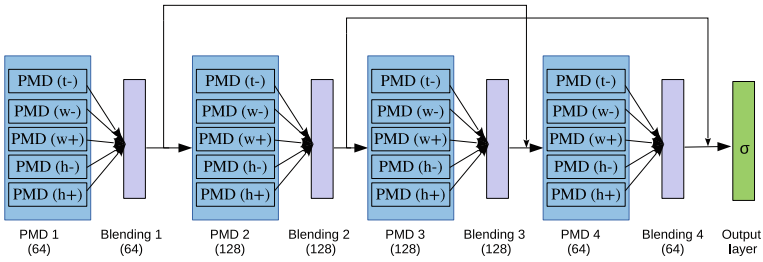

Fig. 4. ContextVP-big (4 layers) architecture: Each layer contains 5 PMD units followed by context blending block. Two skip connections are used, which simply concatenate outputs of two layers (layers 1 & 3, and 2 & 4). The output layer uses the sigmoid activation function which outputs values in the range $(0, 1)$. $(\cdot)$ indicates the number of hidden units in each layer. The ContextVP-small architecture has half the hidden units at each layer.

Network architecture: our best model architecture is illustrated in Fig. 4. It consists of a stack of four context-aware layers with skip connections that directly predicts the scaled RGB values of the next frame. All results are reported for models using 3×3 convolutional kernels for all PMD units, identity activation function in Eqs. 2 and 4 and training using $\mathcal{L}_1$ ($p = 1$ in Eq. 5) with GDL loss. Changing to 5×5 size kernels, use of nonlinear activations (e.g., ReLU [23] or tanh) or layer normalization [1] in the blending blocks does not affect the performance in our experiments. Finn et al. [6] reported that $\mathcal{L}_1$ with the GDL loss function performs better than $\mathcal{L}_2$ but their performance in our case was very similar.

Baseline: our baseline (ConvLSTM20) is a network consisting of a stack of 20 ConvLSTM layers with kernels of size 3×3. The number of layers was chosen

[1] Mean Squared Error (MSE) is also reported for the car-mounted camera video prediction to compare with PredNet [21].

to be 20 to cover a large context and also since each layer in our 4-layer model consists of 5 PMD units. Two skip connections similar to our model were also used. The layer sizes are choen to keep the number parameters comparable to our best model (ContextVP4-WD-big). Surprisingly, **this baseline outperforms almost all state of the art models** except Deep Voxel Flow [20] on the UCF-101 dataset. Note that it is **less amenable to parallelization** compared to ContextVP models where PMD units for different directions can be applied in parallel.

Table 1. Results of ablation study on the Human3.6M dataset. The model is trained on 10 frames and predicts the next frame. The results are averaged over test videos. D indicates directional weight-sharing, and U and W indicate uniform and weighted blending, respectively. Higher values of PSNR/SSIM and lower values of MSE indicate better results.

Name	# layers	Blending type	DWS	PSNR	SSIM	# parameters
ContextVP1	1	Uniform (U)	N	38.1	0.990	0.7 M
ContextVP3	3	Uniform (U)	N	41.2	0.992	1.6 M
ContextVP4-U-big	4	Uniform (U)	N	42.3	0.994	14.0 M
ContextVP4-W-big	4	Weighted (W)	N	44.8	0.996	14.2 M
ContextVP4-WD-small	4	Weighted (W)	Y	45.0	0.996	2.0 M
ContextVP4-WD-big	4	Weighted (W)	Y	**45.2**	**0.996**	8.6 M

Table 2. Evaluation of Next-Frame Predictions on the Human3.6M dataset. All models are trained on 10 frames and predicts the next frame. The results are averaged over test videos.ConvLSTM20 is our baseline containing 20 ConvLSTM layers. Higher values of PSNR and SSIM, lower values of MSE indicate better results. Our best models (ContextVP4-WD: 4 layers with weighted blending and DWS) outperform our baseline as well as current state-of-the-art methods with fewer number of parameters.

Method	PSNR	SSIM	# parameters	Time (s)
Copy-Last-Frame	32	-	-	-
BeyondMSE [22]	26.7	-	8.9 M	-
PredNet [21]	38.9	-	6.9 M	-
ConvLSTM20	44.1	0.995	9.0 M	0.153
ContextVP4-WD-small	45.0	0.996	2.0 M	-
ContextVP4-WD-big	**45.2**	**0.996**	8.6 M	**0.092**

5.1 Human Motion Prediction (Human3.6M dataset)

We first evaluate our model on Human3.6M dataset [15]. The dataset includes seven human subjects (three females and four males). Five subjects are used for training and the other two for validation and testing. The videos are subsampled to 10 fps and downsampled to 64×64 resolution.

Ablation study: using this dataset, we evaluate the importance of various components of our model: multiple layers, types of context blending, and DWS regularization. Table 1 shows the results. We find that performance increases substantially with number of layers, switch to W-blending, and addition of DWS. ContextVP1, ContextVP3 and ContextVP4-U-big use U-blending and no DWS, corresponding to direct adaptation of PyraMiD-LSTM for video prediction.

Comparison to other methods: Table 2 shows the comparison of the prediction results with the Baseline ConvLSTM as well as PredNet [21], and BeyondMSE [22]. Another baseline Copy-Last-Frame is included to show the result of simply copying the last input frame. We do not compare to Finn et al. [6] since their model was not trained for next-frame prediction. From Table 1, it can be seen that single layer ContextVP already outperforms BeyondMSE which uses 3D-CNN, and the three-layer ContextVP networks outperform PredNet which uses ConvLSTM. Finally, four layer ContextVP networks with W-blending and DWS outperform all approaches, even with much fewer parameters (ContextVP4-WD-small). Increasing the model size (ContextVP4-WD-big) brings a minor improvement in final performance.

Table 3. Evaluation of Next frame prediction on the CalTech Pedestrian dataset (trained on the KITTI dataset). All models are trained on 10 frames and predicts the next frame. The results are averaged over test videos. ConvLSTM20 is our baseline containing 20 ConvLSTM layers. Higher values of PSNR and SSIM, lower values of MSE indicate better results. (+) This score is provided by [19]. (*) The scores provided in Lotter et al. [21] are averaged over nine frames (time steps 2–10 in their study), but ours are computed only on the next predicted frame. We therefore re-calculated the scores of PredNet using their trained network. Our best models (ContextVP4-WD: 4 layers with weighted blending and DWS) outperform the baseline as well as current state-of-the-art methods with fewer number of parameters.

Method	MSE ($\times 10^{-3}$)	PSNR	SSIM	# parameters	Time (s)
Copy-Last-Frame	7.95	23.3	0.779	-	-
+BeyondMSE [22]	3.26	-	0.881	-	-
*PredNet [21]	2.42	27.6	0.905	6.9 M	-
Dual Motion GAN [19]	2.41	-	0.899	113 M	-
ConvLSTM20	2.26	28.0	0.913	9.0 M	0.447
ContextVP4-WD-small	2.11	28.2	0.912	2.0 M	-
ContextVP4-WD-big	**1.94**	**28.7**	**0.921**	8.6 M	**0.346**

Fig. 5. Qualitative comparisons from the test set among our best model (ContextVP4-WD-big), the baseline (ConvLSTM20), and the state-of-the-art model (PredNet). All models are trained for next-frame prediction given 10 input frames on the KITTI dataset, and tested on the CalTech Pedestrian dataset.

5.2 Car-Mounted Camera Video Prediction (KITTI and CalTech Pedestrian Dataset)

The model is trained on the KITTI dataset [7] and tested on the CalTech Pedestrian dataset [5]. Every ten input frames from "City", "Residential", and "Road" videos are sampled for training resulting in ≈41 K frames. Frames from both datasets are center-cropped and down-sampled to 128×160 pixels. We use the exact data preparation as PredNet [21] for direct comparison.

The car-mounted camera videos are taken from moving vehicles and consist of a wide range of motions. Compared to Human3.6M, which has static background and small motion flow, this dataset has diverse and large motion of cars at different scales and also has large camera movements. To make predictions for such videos, a model is required to learn not only small movement of pedestrians, but also relatively large motion of surrounding vehicles and backgrounds.

We compare our approach with the Copy-Last-Frame and ConvLSTM20 baselines as well as BeyondMSE, PredNet, and Dual Motion GAN [19] which are the current best models for this dataset. Note that the scores provided in Lotter et al. [21] are averaged over nine frames (time steps 2–10 in their study), but ours are computed only on the next predicted frame. Therefore, we re-calculated the scores of PredNet for the next frame using their trained network. As shown in Table 3, our four layer model with W-blending and DWS outperforms the state-of-the-art on all metrics. Once again, the smaller ContextVP network already matches the baseline while being much smaller and more suitable for paral-

lelization. Some samples of the prediction results from the test set are provided in FIg. 5. Our model is able to adapt predictions to the current scene and make sharper predictions compared to the baseline and PredNet.

Table 4. Evaluation of Next-Frame Predictions on the UCF-101 dataset. Models are trained on four frames and predict the next frame. Results are averaged over test videos. ConvLSTM20 is our baseline containing 20 ConvLSTM layers. (∗) Liu et al. [20] did not provide the number of parameters but noted that their model has the same number of parameters as BeyondMSE [22]. Higher values of PSNR and SSIM, lower values of MSE indicate better results. For UCF-101 dataset, larger kernel size produces the better prediction using fewer number of parameters. Our best models (ContextVP4-WD: 4 layers with weighted blending and DWS) outperform the baseline as well as current state-of-the-art methods with fewer number of parameters.

Method	PSNR	SSIM	# parameters	Time (s)
BeyondMSE [22]	32	0.92	8.9 M	-
MCnet+RES [34]	31	0.91	14 M	-
DVF [20]	33.4	**0.94**	≈8.9 M∗	-
ConvLSTM20	32.9	0.91	9.0 M	0.499
ContextVP4-WD-small	34.7	0.92	2 M	-
ContextVP4-WD-big	**34.9**	0.92	8.6 M	**0.474**

5.3 Human Action Prediction (UCF-101 Dataset)

The last dataset we test on is UCF-101 [29] consisting of videos from YouTube. Although many videos in this dataset contain small movements between frames, they contain much more diversity in objects, backgrounds and camera motions compared to previous datasets. Our experimental setup follows that of Mathieu et al. [22]. About 500 K training videos are selected from the UCF-101 training set, and 10% of UCF-101 test set is used for testing (378 videos). All frames are resized to 256×256. Note that Mathieu et al. used randomly selected sequences of 32×32 patches from the Sports-1M dataset [17] for training since the motion between frames in the UCF-101 dataset are too small to learn dynamics. Our model however, is directly trained on UCF-101 subsequences of length four with the original resolution. Motion masks generated using Epicflow [27] provided by Mathieu et al. are used for validation and testing, so the evaluation is focused on regions with significant motion when computing PSNR and SSIM.

Table 4 presents the quantitative comparison to the baseline as well as four best results from past work: adversarial training (BeyondMSE; [22]), the best model from Villegas et al. (MCnet+RES; [34]), and Deep Voxel Flow (DVF; [20]). The results are similar to previous datasets: even at much smaller size, the four layer ContextVP network outperforms the baseline and other methods, and increasing the model size brings a small improvement in score. However, it does

not outperform DVF on SSIM score. These results shows that small ContextVP models can capture relevant spatial-temporal information and use it to make sharp predictions in very diverse settings without requiring adversarial training.

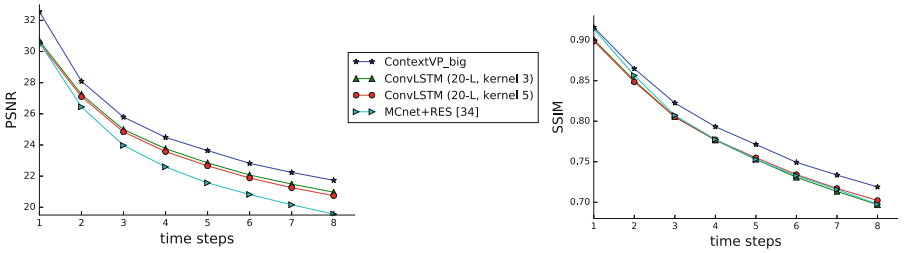

Fig. 6. Comparison of multi-step prediction on UCF101: our best models (ContextVP), Villegas et al. (MCnet+RES; [34]), and 20-layer ConvLSTM baseline. Given 4 input frames, the models are trained for next-frame prediction and tested to predict 8 frames recursively.

Multi-Step Prediction: Figure 6 compares multi-step prediction results of our models with the baseline (ConvLSTM20), MCnet+RES, and BeyondMSE. Given four frames, all networks were trained for single frame prediction and scored on the test set by predicting eight frame recursively. Our small and big models perform very similarly according to PSNR, but the SSIM score for further predictions are better for the *smaller* model. Qualitative comparisons are presented in Fig. 7. In the first video, ContextVP produces clear predictions for the subject's face and fewer motion artifacts for the black object, as opposed to other methods. In the second video, more details of the rider and the horse are preserved by ContextVP.

6 Conclusion and Future Directions

This paper identified the issue of missing context in current video prediction models, which contributes to uncertain predictions about the future and leads to generation of blurry frames. To address this issue, we developed a novel prediction architecture that captures all of the relevant context efficiently at each layer. It outperformed existing approaches for video prediction in a variety of scenarios, demonstrating the importance of fully context-aware models.

We did not incorporate other recent ideas for improve video prediction such as explicit background/motion flow modeling, or adversarial training. Since these have been previously explored for models with incomplete context, a promising future direction is to evaluate their influence on fully context-aware models. Our work suggests that full context coverage should be a required feature of any video prediction baseline to rule out multiple sources of uncertainty.

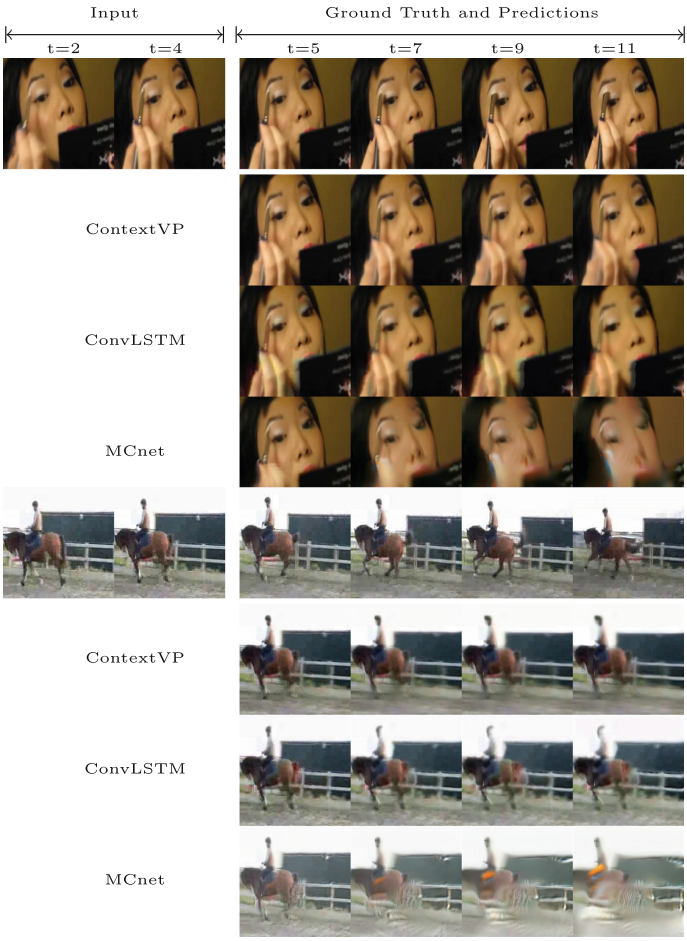

Fig. 7. Qualitative comparisons from the UCF-101 test set among our best model (ContextVP4-WD-big), the baseline (ConvLSTM20), and the state-of-the-art model (MCNet). All models are trained for next-frame prediction given 4 input frames. They are then tested to recursively predict 8 future frames (see also Fig. 6).

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. Baldi, P., Pollastri, G.: The principled design of large-scale recursive neural network architectures-dag-rnns and the protein structure prediction problem. J. Mach. Learn. Res. **4**, 575–602 (2003)
3. Byeon, W., Breuel, T.M., Raue, F., Liwicki, M.: Scene labeling with lstm recurrent neural networks. In: CVPR (2015)

4. Byeon, W., Liwicki, M., Breuel, T.M.: Texture classification using 2d lstm networks. In: 2014 22nd International Conference on Pattern Recognition (ICPR), pp. 1144–1149. IEEE (2014)
5. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: a benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 304–311. IEEE (2009)
6. Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction. In: Advances in Neural Information Processing Systems, pp. 64–72 (2016)
7. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the kitti dataset. Int. J. Robot. Res. **32**(11), 1231–1237 (2013)
8. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. Aistats **9**, 249–256 (2010)
9. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
10. Graves, A., Fernández, S., Schmidhuber, J.: Multi-dimensional recurrent neural networks. In: Proceedings of the 17th International Conference on Artificial Neural Networks, September 2007
11. Graves, A., Schmidhuber, J.: Offline handwriting recognition with multidimensional recurrent neural networks. In: NIPS (2009)
12. Graves, A., Fernández, S., Schmidhuber, J.: Multi-dimensional recurrent neural networks. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D. (eds.) ICANN 2007. LNCS, vol. 4668, pp. 549–558. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74690-4_56
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
14. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1325–1339 (2014)
15. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE Trans.n Pattern Anal. Mach. Intell. **36**(7), 1325–1339 (2014)
16. Kalchbrenner, N., et al.: Video pixel networks. arXiv preprint arXiv:1610.00527 (2016)
17. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
18. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Liang, X., Lee, L., Dai, W., Xing, E.P.: Dual motion gan for future-flow embedded video prediction. arXiv preprint arXiv:1708.00284 (2017)
20. Liu, Z., Yeh, R., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. arXiv preprint arXiv:1702.02463 (2017)
21. Lotter, W., Kreiman, G., Cox, D.: Deep predictive coding networks for video prediction and unsupervised learning. arXiv preprint arXiv:1605.08104 (2016)
22. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. arXiv preprint arXiv:1511.05440 (2015)
23. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp. 807–814 (2010)

24. Oh, J., Guo, X., Lee, H., Lewis, R.L., Singh, S.: Action-conditional video prediction using deep networks in atari games. In: Advances in Neural Information Processing Systems, pp. 2863–2871 (2015)
25. Patraucean, V., Handa, A., Cipolla, R.: Spatio-temporal video autoencoder with differentiable memory. arXiv preprint arXiv:1511.06309 (2015)
26. Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., Chopra, S.: Video (language) modeling: a baseline for generative models of natural videos. arXiv preprint arXiv:1412.6604 (2014)
27. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: EpicFlow: edge-preserving interpolation of correspondences for optical flow. In: Computer Vision and Pattern Recognition (2015)
28. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, pp. 568–576 (2014)
29. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: a dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
30. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: International Conference on Machine Learning, pp. 843–852 (2015)
31. Stollenga, M.F., Byeon, W., Liwicki, M., Schmidhuber, J.: Parallel multi-dimensional lstm, with application to fast biomedical volumetric image segmentation. In: Advances in Neural Information Processing Systems, pp. 2998–3006 (2015)
32. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
33. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: C3d: generic features for video analysis. CoRR, abs/1412.0767 2, 7 (2014)
34. Villegas, R., et al.: Decomposing motion and content for natural video sequence prediction. In: ICLR (2017)
35. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: Advances in Neural Information Processing Systems, pp. 613–621 (2016)
36. Wang, Y., Long, M., Wang, J., Gao, Z., Philip, S.Y.: Predrnn: recurrent neural networks for predictive learning using spatiotemporal lstms. In: Advances in Neural Information Processing Systems, pp. 879–888 (2017)
37. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
38. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C.: Convolutional lstm network: a machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems, pp. 802–810 (2015)
39. Xue, T., Wu, J., Bouman, K., Freeman, B.: Visual dynamics: probabilistic future frame synthesis via cross convolutional networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2016)
40. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)