One-Step Time-Dependent Future Video Frame Prediction with a Convolutional Encoder-Decoder Neural Network

Vedran Vukotić, Silvia-Laura Pintea, Christian Raymond, Guillaume Gravier, Jan van Gemert

One-Step Time-Dependent Future Video Frame Prediction with a Convolutional Encoder-Decoder Neural Network

Vedran Vukotić[1,2,3], *Silvia-Laura Pintea*[1], *Christian Raymond*[2,3], *Guillaume Gravier*[2,4], *Jan van Gemert*[1]

[1]TU Delft, Delft, The Netherlands
[2]INRIA/IRISA, Rennes, France
[3]INSA Rennes, Rennes, France
[4]CNRS, France

{vedran.vukotic, christian.raymond, guillaume.gravier}@irisa.fr
{S.L.Pintea, j.c.vangemert}@tudelft.nl

Abstract

There is an inherent need for machines to have a notion of how entities within their environment behave and to anticipate changes in the near future. In this work, we focus on anticipating future appearance, given the current frame of a video. Typical methods are used either to predict the next frame of a video or to predict future optical flow or trajectories based on a single video frame. This work presents an experiment on stretching the ability of CNNs to anticipate appearance at an arbitrarily given near future time, by conditioning our predicted video frames on a continuous time variable. We show that CNNs can learn an intrinsic representation of typical appearance changes over time and successfully generate realistic predictions in one step - at a deliberate time difference in the near future. The method is evaluated on the KTH human actions dataset and compared to a baseline consisting of an analogous CNN architecture that is not time-aware.

1. Introduction

For machines to be able to successfully interact in real world scenarios, they would need to be able to anticipate actions and events and plan accordingly. This is however a difficult task since even with recent advances in deep and reinforcement learning, machines still do not posses complex knowledge of the world and are rather adapted to specific narrow tasks. If we limit the task to anticipating future appearance, machines have a slight advantage due to the vast collection of unlabeled videos available today which is perfectly suited for unsupervised learning methods. To anticipate future appearance based on current visual information, a machine needs to successfully be able to recognize entities and their parts, as well as to develop an internal representation of how does the movement happen in regards to time.

We start from a given input video frame and aim to predict a future video frame at a given temporal distance, Δt away from the input frame. We achieve this by conditioning our video frame prediction on an input time-indicating variable and we are able to predict a future video frame that is temporally further away from the input given frame in one step. Therefore, in this work we propose one-step, long-term video frame prediction. This is beneficial both in terms of computational efficiency, as well as not having to concern with the propagation and accumulation of prediction errors, as in the case of sequential/iterative prediction.

Our work falls into the autoencoding category, where a current image is presented as input and an image resembling the anticipated future is provided as output. Most typically, such models are trained to predict a frame that is Δt in the future, while anticipations that are further away are predicted in an iterative way. Our proposed method consists of an encoding CNN, a decoding CNN and a separate branch, parallel to the encoder, that models time.

Machines typically have a response time $\Delta t_{response}$. Being able to anticipate the near future allows them to correct for their inherent delay and to plan accordingly. Anticipating the near future is especially useful in robotics, where artificial systems have to interact with their environment in real time.

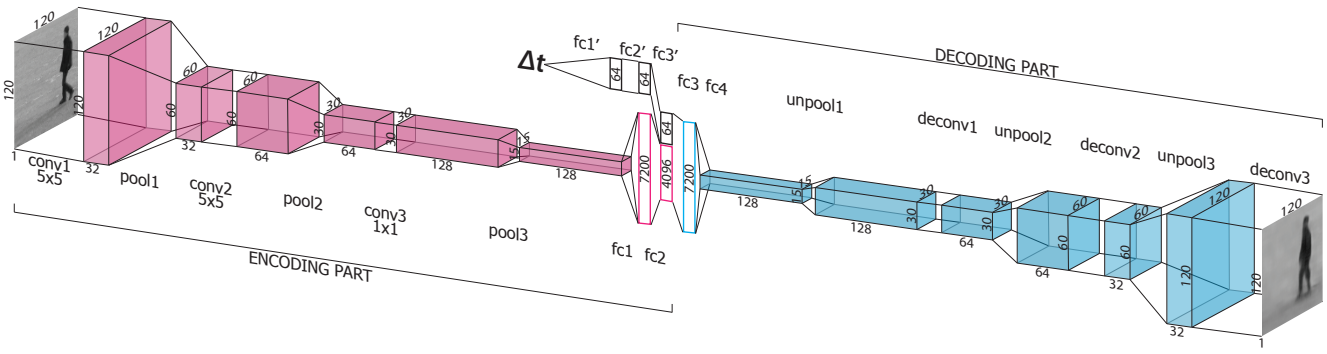

Figure 1. Our proposed architecture consists of two parts: i) an encoder part consisting of two branches: the first one taking the current image and the second one taking an arbitrary time difference t to the desired prediction and ii) a decoder part that generates an image, as anticipated, at the desired input time difference.

1.1. Related Work

1.1.1 Predicting Future Actions

In the context of action prediction, it has been shown that it is possible to use high level embeddings to anticipate future actions up to one second before they begin [24]. Predicting the future event by retrieving similar videos and transferring this information, is proposed in [29]. In [9] a hierarchical representation is used for predicting future actions. Predicting a future activity based on analyzing object trajectories is proposed in [7]. In [4], the authors forecast human interaction by relying on body-pose trajectories. In the context of robotics, in [8] human activities are anticipated by considering the object affordances. Unlike these works, rather than predicting actions, we focus on predicting a single video frame at a given future temporal displacement from a given input video frame.

1.1.2 Predicting Future Motion

Anticipating future movement in the spatial domain as closely as possible to the real movement has also been previously considered. For this case, the methods start from an input image at the current time stamp and predict OF (Optical Flow) at the next timestep. In [10] images are aligned to their nearest neighbour in a database and the motion prediction is obtained by transferring the motion from the nearest neighbor to the input image. In [13], structured random forests are used to predict OF vectors at the next timestep. In [12], the use of LSTM is advised towards predicting Eulerian future motion. A custom deep convolutional neural network is proposed in [28] towards future OF prediction. Rather than predicting the motion at a future moment in time, in [26] the authors propose to predict motion trajectories using variational autoencoders. This is similar to predicting OF, but given the temporal consistency of the trajectories it offers greater accuracy. Dissimilar to these methods, we aim to predict the video appearance information at a given future temporal displacement, from an input video frame. Predicting appearance rather than motion is beneficial as the predicted outcome is spatially coherent.

Focusing on the object motion as given by their dynamics in real world, is proposed in [11], by relying on Newtonian physical laws. In [2], the future location of objects is predicted by learning from synthetic abstract data. This can be seen somewhat related to learning to predict OF, which is also an indicator of displacement. Unlike these methods, we aim to predict the appearance of a future video frame given an input video frame and a desired temporal difference.

1.1.3 Predicting Future Appearance

One intuitive trend towards predicting future information is predicting future appearance. In [27], the authors propose to predict both appearance and motion for street scenes using top cameras. Predicting patch-based future video appearance, is proposed in [15], by relying on large visual dictionaries. Similar to these methods, we also aim at predicting the appearance of future video frames, however we condition our prediction on a time parameter than allows us to perform the prediction efficiently, in one step.

More recent methods rely on convolutional neural networks towards predicting possible video frames. Rather than predicting future appearance from input appearance information, hallucinating possible images has been a recent focus. The novel work in [25] relies on the generative adversarial network model [14] to create not only the appearance of an image but also the possible future motion. This is done using spatio-temporal convolutions that discriminate between foreground and background. Similarly, in [18] a temporal generative neural network is proposed towards generating more robust videos. These generative models can be conditioned to generate feasible outputs given a specific conditioning input [16, 26]. Dissimilar to them, we rely on an autoencoding model. Autoencoding methods try to encode the current image in a representation space that

is suitable for learning appearance and motion, and decode such representations to retrieve the anticipated future, either as an image or optical flow/trajectories. Here, we propose to use video frame appearance towards predicting future video frames, conditioned on a given time indicator.

Related to predicting future appearance, the recent works in [22, 23] propose predicting future image pixels conditioned on all previous seen pixels — possible image completions from a set of initial pixels. Unlike these methods, we aim to predict complete future images from a provided input image and a provided temporal displacement.

Similar to transferring the optical flow vectors between images, as considered in [10], appearance transfer has also been considered. Works such as [3, 5, 17] focus on the task of artistic style transfer from a given input image to another image or video. Unlike these methods, we do not transfer a given appearance but rather predict a future frame appearance. We do so by conditioning on a parameter indicating the desired time displacement between the input frame and the predicted frame.

2. Time-dependent Video Frame Prediction

To tackle the problem of anticipating future appearance at arbitrary temporal distances we deploy an encoder-decoder architecture. The encoder has two separate branches, one to receive the input image, and one to receive the desired temporal displacement Δt of the prediction. The decoder than takes the input from the encoder and generates a feasable prediction for the given input image and the desired temporal displacement. This is illustrated in Figure 1. The network receives as inputs an image and a variable Δt, $\Delta t \in \mathbb{R}^+$, indicating the time difference from the time of the provided image to the time of the desired prediction. The network predicts an image at the anticipated future time $t_0 + \Delta t$. We use a similar architecture to the one proposed in [21]. However, while their architecture is made to encode RGB images and a continuous angle variable to produce RGBD as output, our architecture is designed to take as input a monochromatic image and a continuous time variable, Δt, and to produce a monochromatic image as output. More specifically, the architecture consist of the following:

- *an encoding part* composed of two branches:
  - *an image encoding branch* defined by 4 convolutional layers, 3 pooling layers and 2 fully-connected layers at the end;
  - *a time encoding branch* consisting of 3 fully-connected layers.

  The final layers of the two branches are concatenated together, forming one bigger layer that is then provided to the decoding part.
- *a decoding part* composed of 2 fully-connected layers, 3 "unpooling" (upscaling) layers, and 3 "deconvolutional" (transpose convolutional) layers.

The input time variable is continuous and allows for appearance anticipations at arbitrary time differences. Possible alternatives of the proposed architecture could include encoded time inputs (*e.g.* multiple input neurons) or a continuous time variable followed by an embedding layer (*e.g.* lookup table). The downside of these approaches would be the discretization of the time input.

2.1. Network Training

Training is performed by presenting batches of $\{I_x, \Delta t, I_y\}$ tuples, where I_x represents an input image at current relative time t_0, Δt represents a continuous variable indicating the time difference to the future video frame and I_y represents the actual video frame at $t_0 + \Delta t$.

2.2. Network Prediction

Predictions are obtained in one step. For every input image I_x and continuous time difference variable Δt, a $\{I, \Delta t\}$ pair is given to the network and an image representing the appearance anticipation I_y after a time interval Δt is directly obtained as output. No iterative steps are performed.

3. Experiments

3.1. Experimental Setup

We evaluate our method by generating multiple images of anticipated future appearances and comparing them both visually and through MSE (Mean Squared Error) with the true future frames, as well as to a CNN baseline method that sequentially predicts the future video frame. For the baseline method, we use a CNN encoder-decoder architecture that does not have a notion of time and is used in an iterative manner to produce anticipated futures at $k\Delta t$ $(k = 1, 2, ...)$ temporal displacements.

To test the architecture proposed in Section 2, we implemented it by using the *TensorFlow* [1] framework. We use the Adam optimizer [6], with L_2 loss and dropout rate set to 80% for training. Training is performed up to 500000 epochs with randomized minibatches consisting of 16 samples where each sample contains one input image at current relative time $t_0 = 0$, a temporal displacement Δt $(\Delta t < 200ms)$ and the real frame at the desired temporal displacement Δt. On a *Titan X* GPU, training took approximately 16 hours with, on average, about 100,000 training samples (varying in each action category). It is important to note that introducing sparsity in the central layers did not improved the results, but rather deteriorate them so we opt not to introduce sparsity. We do not use early stopping and

we ran each experiment for the full number of epochs. We argue that the type of action can be easily automatically detected and is better incorporated by training a network per action category. Thus, we opt to perform separate preliminary experiments for each action instead of training one heavy network to anticipate video frames corresponding to all the different possible actions.

3.1.1 Encoder Architecture

Given that the input, and thus also the output, image size is $120 \times 120 \times 1$ (120×120 grayscale images), in our encoder part, we stack convolutional and pooling layers that yield consecutive feature maps of the following decreasing sizes: 120×120, 60×60, 30×30 and 15×15 with an increasing number of feature maps per layer, namely 32, 64 and 128 respectively. Fully-connected layers of sizes 7200 and 4096 follow. The separated branch of the encoder that models time consists of 4 fully connected layers of size 64, where the last layer is concatenated to the fully-connected layer on top of the convolutional neural networks. This yields an embedding of size 4160 that is presented to the decoder. Kernel sizes used for the convolutional operations start at 5×5 in the first layers and decrease to 2×2 and 1×1 in the deeper layers of the encoder. For the decoder, the kernel sizes are ordered in the opposite direction.

3.1.2 Decoder Architecture

The decoder consists of interchanging "unpooling" (upscaling) and "deconvolutiton" (transpose convolution) layers, yielding feature maps of the same sizes as the image-encoding branch of the encoder, only in the opposing direction. For simplicity, we implement pooling as a convolution with 2×2 strips and unpooling as a 2D transpose convolution. It is worth noting that sometimes pooling/unpooling layers are completely omitted [20, 21] in similar encoder-decoder CNN architectures with no significant impact on performance. We decided to keep them as a regularization term given that our input and output images differ less and have a more similar appearance than in the case of rotated images [21].

3.2. Dataset

We use the KTH human action recognition dataset [19] for evaluating our proposed method. The dataset consists of 6 different human actions, namely walking, jogging, running, hand-clapping, hand-waving and boxing. Each action is performed by 25 actors. There are 4 video recordings for each action performed by each actor. Inside every video recording, the action is performed multiple times and information about the time when each action starts starts and ends is provided with the dataset.

To evaluate our proposed method properly, we randomly split the dataset by actors, in a training set — with 80% of the actors, and a testing set — with 20% of the actors. By doing so, we ensure that no actor is present in both the training and the testing split and that the network can generalize well with different looking people and does not overfit to specific characteristics of specific actors. The dataset provides video sections of each motion in different directions — *e.g.* walking from right to left and from left to right. This provides a good setup to check if the network is able to understand human poses and locations, and correctly anticipate the direction of movement. The dataset was processed as follows: frames of original size 160×120 were cropped to 120×120 and the starting/ending times of each action are adjusted accordingly to match the new cropped area. Time was estimated based on the video framerate and the respective frame numbers.

3.3. Experimental Results

Our method is evaluated as follows: an image at a considered time, $t_0 = 0$ and a time difference Δt is given as input. The provided output represents the anticipated future frame at time $t_0 + \Delta t$, where Δt represents the number of milliseconds after the provided image.

The sequential encoder-decoder baseline method is evaluated by presenting solely an image, considered at time $t_0 = 0$ and expecting an image anticipating the future at $t_0 + \Delta t_b$ as output. This image is then fed back into the network in order to produce an anticipation of the future at time $t_0 + k\Delta t_b$, $k = 1, 2, 3, ...$

For simplicity, we consider $t_0 = 0ms$ and refer to Δt as simply t. It is important to note that our method models time as a continuous variable. This enables the model to predict future appearances at previously unseen time intervals, as seen in Figure 5. The model is trained on temporal displacements defined by the framerate of the training videos. Due to the continuity of the temporal variable, it can successfully generate predictions for: i) temporal displacements found in the videos (*e.g.* *t*={*40ms, 80ms, 120ms, 160ms, 200ms*}, ii) unseen temporal displacement within the the values found in the training videos (*e.g.* *t*={*60ms, 100ms, 140ms, 180ms*}) and iii) unseen temporal displacement after the maximal value encounter during training (*e.g.* *t=220ms*).

Since both the baseline method and the groundtruth are quantized by the video framerate, the images displayed in Figure 2 are all images at intervals of 40 ms (derived from a framerate of 25fps) for a fair and exact comparison. Figure 2 a) illustrates the case of a person moving from approximately right to left, from the camera viewpoint, at walking speed. Despite the blurring, especially around the left leg when asked to predict for $t = 120ms$, it can be noticed that the our proposed network correctly estimated the location

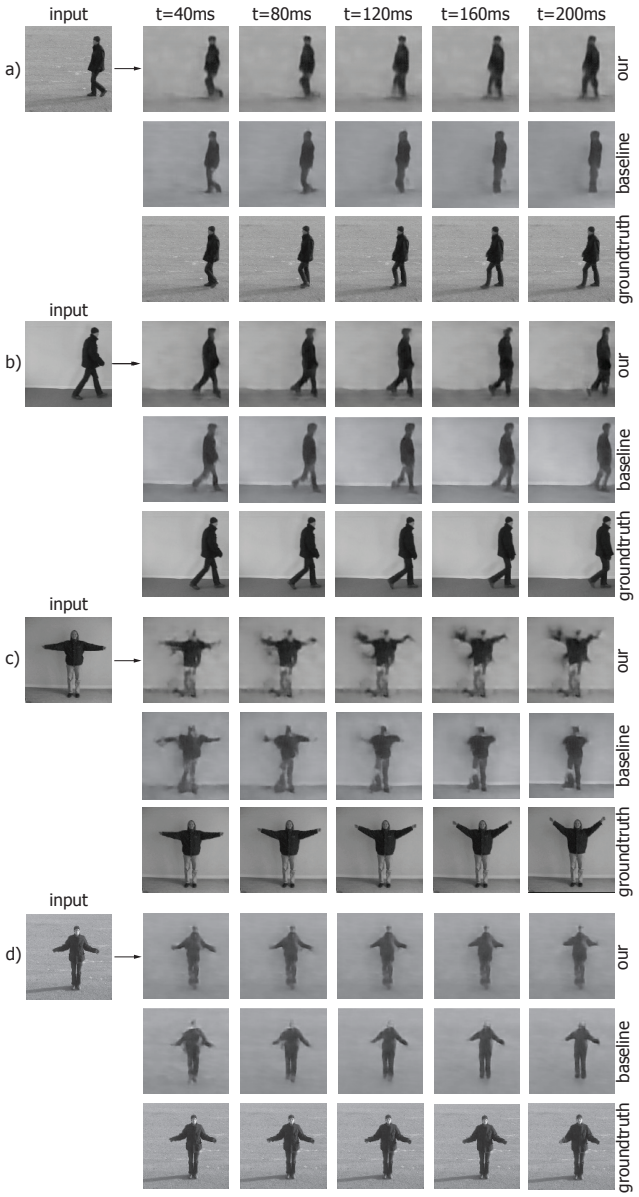

Figure 2. Comparison of predictions for a) a person walking to the left, b) a person walking to the right, c) a person waving with their hands and d) a person slowly clapping with their hands. Given an input picture (on the left) and a time interval (different columns) anticipated future motions are presented for our proposed method and for the baseline convolutional encoder-decoder. The third set of images in each group present the actual future — the groundtruth.

of the person and positioning of body parts. For each time difference, the body-part predictions are realistic, as well as the displacement of the whole person, which matches the groundtruth displacement.

Figure 2 b) again illustrates a person walking, this time approximately left to right. Our proposed network correctly localized the person and the body parts. The network is able to estimate the body pose and thus the direction of movement. Our network correctly predicts the displacement of the person to the right for any given time difference, from just the single input image. The network is able to capture the characteristics of the human gait, as it predicts correctly the alternation in the position of the legs. The anticipated future frame is realistic but not always perfect, as it is hard to perfectly estimate walking velocity solely from one static image. This can be seen at $t = 200ms$ in Figure 2 b). Our network predicts one leg to further behind while the actor, as seen in the groundtruth, was moving slightly faster and moved the leg past the knee of the other leg.

Our proposed network is able to learn an internal representation that is capable of encoding the stance of the person such that it correctly predicts the location of the person, as well as to anticipate their new body pose after a deliberate temporal displacement. The baseline network does not have a notion of time and therefore relies on iterative predictions. Time is quantized and the network is trained to generate an anticipated image at time $t + \Delta t$, given an image at time $t = 0$. After that, the process is repeated iteratively, which affects the performance. Figure 2 shows that the baseline network looses the ability to correctly anticipate body movement after some time. This can be best seen in Figure 2 a) where the baseline network correctly predicts the position of the legs up to $t = 80ms$. After that, the network predicts correctly the global dislacement of the person in the correct direction, but body part movements are not anticipated correctly. At $t > 160ms$ the baseline encoder-decoder network shows a big loss of details, enough to cause its inability to correctly model body movement. Therefore, it displays fused legs where they should be separated, as part of the next step the actor is making. Our proposed architecture correctly models both global person displacement and body pose, even at $t = 200ms$.

Figure 2 c) displays an actor handwaving. The proposed network successfully predicts upward movement of the arms and generates images accordingly. In this case however, more artifacts are noticeable. The bidirectional motion of hands during handwaving is ambiguous, as the hand pose does not affect other body parts such as head positioning, or legs. It is important to note that although every future anticipation is independent from each other they are all consistent: *i.e.* it does not happen that the network predicts one movement for t_1 and a different movement for t_2 that is inconsistent with it. This is a strong indicator that the network learns an embedding of appearance changes over time, the necessary filters to react to relevant image areas, and to synthesize correct future anticipations.

However, our proposed model is limited by the total temporal displacement t. For very large time displacements, we expect our frame predictions to deteriorate. This is emphasized in long-term anticipations, as illustrated in Figure 3.

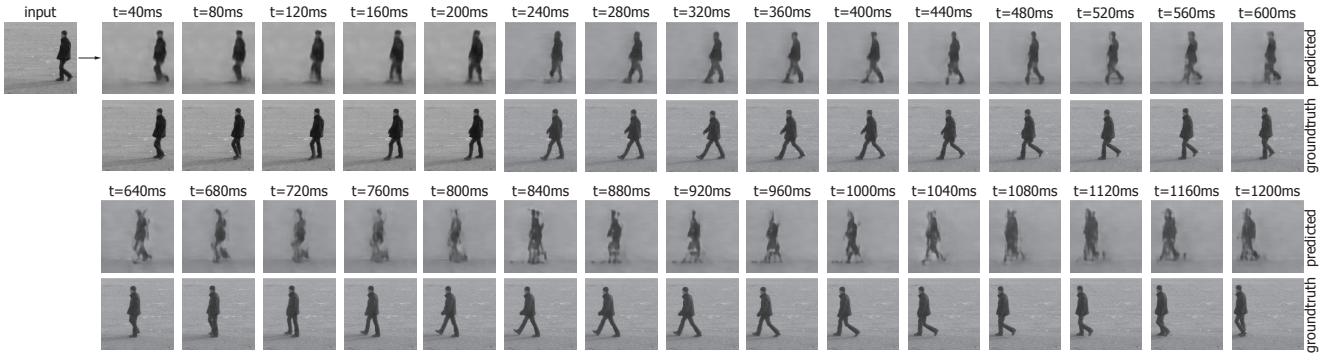

Figure 3. Long distance predictions. For larger temporal displacements artifacting becomes visible. The anticipated location of the person begins to differ from the groundtruth for even larger temporal differences, towards the end of the total motion duration.

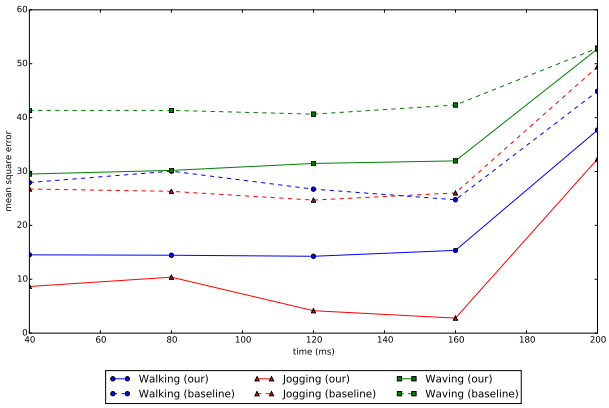

Figure 4. Mean squared error (MSE) over time for certain actions (walking, jogging, waving) for our proposed method and for the convolutional encoder-decoder baseline.

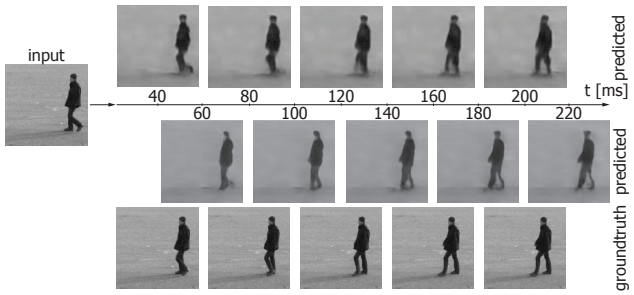

Figure 5. Prediction of seen and unseen temporal displacements. The networks is trained on temporal displacements dictated by the training set video framerate. However, predictions are possible both for seen (1st row, $t = 40 \times k$ ms) and for previously unseen temporal displacements (2nd row, $60 + 40 \times k$ ms).

The smaller the temporal displacement t, the better the prediction is and the lower the MSE score, when compared to the real future frame. In this preliminary study, we do not check the limits of a maximum feasible time difference t, after which our proposed method would provide unsatisfactory results. However, as seen both from the illustrations in Figure 2 and the graphs in Figure 4, our network behaves better with respect to increasing time displacements that the encoder-decoder baseline network. This is supported by the network's ability to predict future video frames at arbitrary future times directly, without having to go through iterative steps that accumulate prediction error.

As expected, not every action is equally challenging for the proposed architecture. Table 1 illustrate MSE scores averaged over multiple time differences, t, and for different predictions from the KTH test set. MSE scores were computed on dilated edges of the groundtruth images to only analyze the part around the person and remove the influence of accumulated variations of the background. A Canny edge detector was used on the groundtruth images. The edges were dilated by 11 pixels and used as a mask for both the groundtruth image and the predicted image. MSE values were computed solely on the masked areas.

We compare our proposed method with the baseline CNN encoder-decoder architecture. It's worth noticing that the MSE does not strictly correlate with qualitative visual inspection. For example, on average, running seems to perform reasonably well, and moreover it outperforms hand-waving, hand-clapping, boxing and even walking. Yet, this is not the case as predictions for running, at the framerate available in the KTH dataset, generate a considerable loss of details and artifacts, as visible in Figure 6 d). These artifacts are not as prominent in the other, less well-performing action categories, in terms of MSE scores.

The average MSE scores, given in Table 1, show that our proposed method outperforms the encoder-decoder CNN baseline by a margin of 13.41, on average, which is expected due to the iterative process of the baseline network.

3.4. Ambiguities and Downsides

As MSE values grouped by different actions indicate, not every action is equally challenging for our proposed method to be anticipated. However, there are a few key factors that make prediction more difficult and cause either the creation

Action	Mean Squared Error	
	Baseline	Our Method
Jogging	30.64	**11.66**
Running	40.88	**17.35**
Walking	30.87	**19.26**
Hand-clapping	43.23	**33.93**
Hand-waving	43.71	**35.19**
Boxing	46.22	**37.71**
Mean MSE	39.26	**25.85**

Table 1. Averaged MSE, over multiple time differences and multiple predictions, on the different action categories of KTH. We compare our method with the baseline convolutional encoder-decoder and show that our method on average performs better than the baseline method in terms of MSE.

of artifacts or loss of details in the generated future frames.

3.4.1 Human Pose Ambiguities

Ambiguities in body-pose happen when the subject is in a pose that does not display inherent information about the movement of the subject in question. A typical example would be when a person is waving, moving their arms up and down, and an image with the arms at a near horizontal position is fed to the network as input. This can results in small artifacts, as visible in Figure 2 c) where for larger time intervals t, although the network is generating upward arm movement, there are visible artifacts that are part of a downward arm movement. A more extreme case is shown in Figure 6 a) where not only does the network predict the movement wrong, upward instead of downward, but it also generates a lot of artifacts with a significant loss of details that increases with the time difference, t.

3.4.2 Fast Movement

Fast movement causes extreme loss of details when the videos provided for training do not offer a high-enough framerate. In other words, this case happens when the visual difference between two consecutive frames during training is substantial — large global displacement and a body pose change that are too large. Examples of this can be seen in Figures 6 b) and c) where the increased speed in jogging and an even more increased speed in running generate significant loss of details. It is important to emphasize that although our proposed architecture can generate predictions at arbitrary time intervals t, the network is still trained on discretized time intervals derived from the videos — intervals that might not be small enough for the network to learn a good motion model. We believe this causes the loss of details and artifacts, and using higher framerate videos during training would alleviate this.

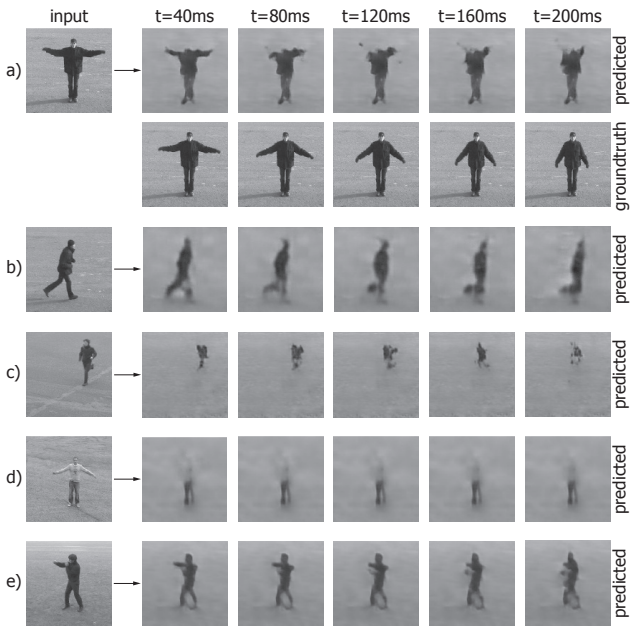

Figure 6. Examples of poor performing future anticipations: a) loss of details in waving, b) loss of details in jogging, c) extreme loss of details in running, d) loss of details with low contrast and e) artifacts in boxing.

3.4.3 Insufficient Foreground/Background Contrast

Decreased contrast between the subject and the background describes a case where the intensity values corresponding to the subject are similar to the ones of the background. This leads to a automatic decrease of MSE values and a more difficult convergence of the network for such cases, which leads to less adaptation and thus to loss of details and artifacts. This can be seen in Figure 6 d). Such effect would be less prominent in case of modeling a network using color images.

3.4.4 Excessive Localization of Movements

Excessive localization of movements happens when the movements of the subject are small and localized. A typical example is provided by the boxing action, as presented in the KTH dataset. Since the hand movement is close to the face and just the hand gets sporadically extended — not a considerable change given the resolution of the images — the network has more difficulties in tackling this. Despite the network predicting feasible movement, often artifacts appear for bigger time intervals t, as visible in Figure 6 e). Although the previously enumerated cases can lead our proposed architecture to predict that display loss of details and artifacts, most can be tackled and removed if necessary by either increasing the framerate, the resolution of the training videos, or using RGB information. The most difficult factor to overcome is human pose ambiguity. We believe this is a

hard problem for our proposed architecture to manage.

4. Conclusion

In this work, we present a convolutional encoder-decoder architecture with a separate input branch that models time in a continuous manner. The aim is to provide anticipations of future video frames for arbitrary positive temporal displacements Δt, given a single image at current time ($t_0 = 0$). We show that such an architecture can successfully learn time-dependant motion representations and synthesize accurate anticipation of future appearance for arbitrary time differences $\Delta t > 0$. We compare our proposed architecture against a baseline consisting of an analogous convolutional encoder-decoder architecture that does not have a notion of time, and show that out method outperforms the baseline both in terms of visual similarity to the groundtruth future video frame, as well as in terms of mean squared error in regards to it. In the last part, we analyze the drawbacks of our architecture and present possible solutions to tackle them. This work shows that convolutional neural networks can inherently model time without having a clear time domain representation. This is a novel notion that can be extended further and that yields high quality anticipations of future video frames for arbitrary temporal displacements, without having to explicitly model the time period in between the provided input video frame and the requested anticipation.

Acknowledgments

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research and the support of GdR ISIS and Rennes Métropole for supporting the collaboration that lead to this research.

References

[1] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 3

[2] D. F. Fouhey and C. L. Zitnick. Predicting object dynamics in scenes. In *CVPR*, pages 2019–2026, 2014. 2

[3] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *CoRR*, 2015. 3

[4] D. A. Huang and K. M. Kitani. Action-reaction: Forecasting the dynamics of human interaction. In *ECCV*, pages 489–504. Springer, 2014. 2

[5] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, 2016. 3

[6] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, 2014. 3

[7] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, pages 201–214. Springer, 2012. 2

[8] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *PAMI*, 38(1):14–29, 2016. 2

[9] T. Lan, T. C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *ECCV*, pages 689–704. Springer, 2014. 2

[10] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 33(5):978–994, 2011. 2, 3

[11] R. Mottaghi, H. Bagherinezhad, M. Rastegari, and A. Farhadi. Newtonian image understanding: Unfolding the dynamics of objects in static images. *CoRR*, 2015. 2

[12] S. L. Pintea and J. C. van Gemert. Making a case for learning motion representations with phase. 2016. 2

[13] S. L. Pintea, J. C. van Gemert, and A. W. M. Smeulders. Déja vu. In *ECCV*, pages 172–187. Springer, 2014. 2

[14] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, 2015. 2

[15] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *CoRR*, 2014. 2

[16] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *CoRR*, 2016. 2

[17] M. Ruder, A. Dosovitskiy, and T. Brox. Artistic style transfer for videos. *CoRR*, 2016. 3

[18] M. Saito and E. Matsumoto. Temporal Generative Adversarial Nets. *CoRR*, 2016. 2

[19] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, volume 3, pages 32–36. IEEE, 2004. 4

[20] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 4

[21] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3d models from single images with a convolutional network. In *ECCV*, pages 322–337. Springer, 2016. 3, 4

[22] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *CoRR*, 2016. 3

[23] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with pixelcnn decoders. *CoRR*, 2016. 3

[24] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *CoRR*, 2015. 2

[25] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, pages 613–621, 2016. 2

[26] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, pages 835–851. Springer, 2016. 2

[27] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. In *CVPR*, pages 3302–3309. IEEE, 2014. 2

[28] J. Walker, A. Gupta, and M. Hebert. Dense optical flow prediction from a static image. In *ICCV*, pages 2443–2451, 2015. 2

[29] J. Yuen and A. Torralba. A data-driven approach for event prediction. In *ECCV*, pages 707–720. Springer, 2010. 2