Hierarchical Model for Long-term Video Prediction

Peter J. Wang
peterwg@berkeley.edu

Zhongxia Yan
zxyan@berkeley.edu

Jeffrey Zhang
jeffzhang1996@berkeley.edu

Abstract

Video prediction has been an active topic of research in the past few years. Many algorithms focus on pixel-level predictions, which generates results that blur and disintegrate within a few frames. In this project, we use a hierarchical approach for long-term video prediction. We aim at estimating high-level structure in the input frame first, then predict how that structure grows in the future. Finally, we use an image analogy network to recover a realistic image from the predicted structure. Our method is largely adopted from the work by Villegas et al.[10] The method is built with a combination of LSTMs and analogy-based convolutional auto-encoder networks. Additionally, in order to generate more realistic frame predictions, we also adopt adversarial loss. We evaluate our method on the Penn Action dataset, and demonstrate good results on high-level long-term structure prediction.

1. Introduction

Learning to predict the future is an important research topic in computer vision and artificial intelligence.[2] Humans make predictions and inferences about the real-world all the time. These predictions focus on the semantics of the action (i.e. where a ball will land, expected overall shape of a figure, etc.), rather than the exact location or pixel intensity. In addition, the rough semantics generally provides more value to the prediction than raw pixel values would anyways.

To better reflect how humans make predictions, we seek to develop a hierarchical approach for video prediction that first predicts some higher-level features of future frames, then regenerate realistic video from the predicted higher-level features. We initially hoped to generate motion segmentations of objects within the videos as the higher-level intermediate features. A recent work by Villegas et al.[10] predicts future poses of humans and then uses image analogies to map the high frequency details of input images to the predicted poses. Since this is very similar to our proposed approach, we decided to first adopt and implement the methods within this paper, then proceed onto replacing pose with motion segmentation in the future.

We evaluate our method on the Penn Action dataset [12], and demonstrate good results on high-level long-term structure prediction.

The rest of the paper is organized as follows: A review of the related work is presented in Section 2. The deep convolutional model we use is presented in Section 3. The experimental results and analysis are shown in Section 4 and 5. Finally we conclude our work with discussion of future work in Section 6.

2. Related Work

There has been numerous recent work on pixel-wise video prediction. Mathieu et al. (2015) [5] uses a multi-scale convolutional generative adversarial network (GAN) approach to recursively generate the next frame given the past few frames. Such pixel-based generations are usually reasonable for the first few predicted frames, but the quality degrades quickly. Vondrick et al. (2016) [11] separately generates a moving foreground representation of a video and a static background and combines the two. Similar approaches include [7], [4] and [9]. Their results, while not photo-realistic, captures the general motion of the foreground object rather well.

The approaches mentioned above use pixel-wise prediction on video frames, but we hope that a hierarchical approach would produce consistent video for a longer duration. Our current hierarchical approach obtains the current pose of the human in the video, predict future poses, and generate realistic video corresponding to the poses. In the future, we will explore replacing pose estimations with motion segmentation, which could generalize to nonhuman objects.

3. Model

3.1. Overview

Our model is a three step process: 1) pose estimation, 2) pose prediction, and 3) image analogy generation. First, given the input frame x_t, our model first get the corresponding pose heatmap p_t via pose estimation. Second, our pose prediction network predicts the pose heatmap for fu-

ture frames. Finally, our image analogy network recovers a realistic image from the corresponding pose heatmap and input frame.

3.2. Pose Estimation

We represent each pose as a list of joint locations. There is work by Zhe Cao et al. [1] and Newell et al. [6] to extract joint locations for different poses. For the purposes of focusing on pose prediction and image analogy, we use the poses annotated by the Penn Action Dataset for our pose predictions.

3.3. Pose Prediction

We create a sequence-to-sequence LSTM network, illustrated in Fig. 1, to predict the future joint locations given a sequence of input joint locations.

$$(h_t, c_t) = LSTM(p_t, h_{t-1}, c_{t-1}),\ 1 \leq t \leq k \quad (1)$$

where $p_t \in \mathbb{R}^{2L}$ is a vector containing the coordinates of all the joint locations at time t, c_t is the LSTM memory cell vector at time t, and h_t is the latent output of the LSTM at time t. Initially, we feed k frames of pose estimation (p_1 to p_k) into the LSTM to give the LSTM context on the video to be predicted.

We generate the future latent outputs h_t for T frames after the k frames of initial input by feeding in $\mathbf{0}$ as the input into the LSTM at each time step.

$$(h_t, c_t) = LSTM(\mathbf{0}, h_{t-1}, c_{t-1}),\ k+1 \leq t \leq k+T \quad (2)$$

We then predict the pose in these T frames by learning a two-layer neural network mapping from the latent output of the LSTM to pose joint locations.

$$\hat{p_t} = \sigma(W_2\sigma(W_1 h_t + b_1) + b_2) \quad (3)$$

where σ is the sigmoid activation function. Notice that we normalized the x and y coordinates in p to be between 0 and 1. Also, we do not feed in the predicted poses $\hat{p}_t$ as input into the LSTM, so that the errors in the predictions does not propagate further into future frames. Instead, this encourages the LSTM model to predict all the future frames only given the original frames.

Our loss function for training the LSTM aims to enforce accuracy of the predictions equally for all T future time points.

$$\mathcal{L}_{LSTM} = \sum_{t=k+1}^{k+T} \|\hat{p}_t - p_t\|^2 \quad (4)$$

3.4. Image Analogy

We use Image Analogy network [8] to synthesize the future frame from its pose structure. As is illustrated in Fig. 2,

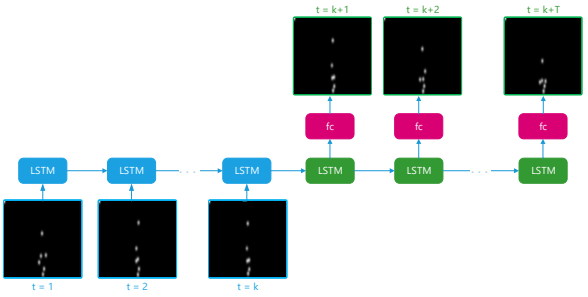


Figure 1. Sequence-to-sequence LSTM model for encoding and decoding the pose structure from video frames.

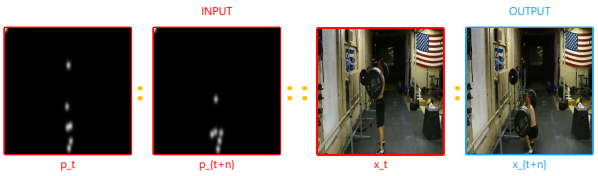


Figure 2. Synthesizing future frame by making analogies between pose structure and image pixels.

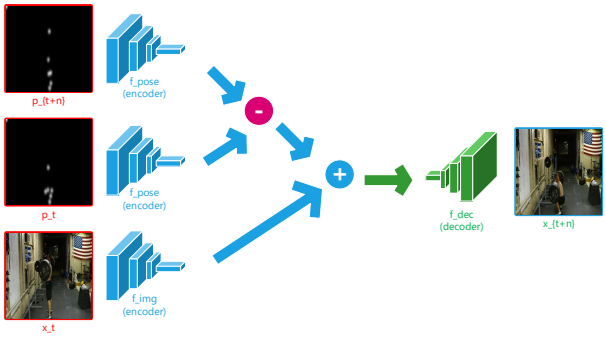


Figure 3. Illustration of our image generator. Our image generator observes an input image, its corresponding human pose, and the human pose of the future image. Through image analogy, our network generates the next frame.

the relationship between p_t and p_{t+n} is the same as the relationship between x_t and x_{t+n}. More specifically, the future frame x_{t+n} can be generated by transferring the structure transformation from p_t to p_{t+n} to the observed frame x_t. To promote more realistic generations, we adopt a generative adversarial network [3] for our image analogy network.

Generator: As is shown in Fig. 3, our image generator uses a pose encoder f_{pose}, an image encoder f_{img} and an image decoder f_{dec}. Specifically, f_{pose} is a convolutional encoder that specializes on identifying key pose features from the pose input that reflects high-level human structure. f_{img} is also a convolutional encoder that acts on an image input by mapping the observed appearance into a feature space where the pose feature transformations can be easily

imposed to synthesize the future frame using the deconvolutional decoder f_{dec}. The image analogy is then performed by

$$\hat{x}_{t+n} = f_{dec}(f_{pose}(g(\hat{p}_{t+n})) - f_{pose}(g(p_t)) + f_{img}(x_t))$$

where $\hat{x}_{t+n}$ and $\hat{p}_{t+n}$ are the generated image and corresponding predicted pose at time $t+n$, x_t and p_t are the input image and corresponding estimated pose at time t, $g(:)$ is a function that maps a set of $(x,\ y)$ joint coordinates to a stack of L heatmaps, and n is the difference in time between the frame to be generated and the input frame.

For the network architecture, our f_{img} encoder is a VGG-16, f_{pose} is a simplified VGG with many layers removed, while f_{dec} is a reversed VGG-16 where all convolutions are replaced by deconvolutions. Our reasoning for removing some VGG-16 hidden layers in f_{pose} is that the pose heatmaps are relatively simple and do not need as much representative power (and also our GPUs are dying).

The loss of generator is written as follows:

$$\mathcal{L}_{generator} = \lambda_{img}\mathcal{L}_{img} + \lambda_{feat}\mathcal{L}_{feat} + \lambda_{adv}\mathcal{L}_{adv}$$

where $\mathcal{L}_{img} = \|x_{t+n} - \hat{x}_{t+n}\|_2^2$ is the ℓ-2 distance between ground-truth and predicted in image space, $\mathcal{L}_{feat} = \|\text{AlexNet}(x_{t+n}) - \text{AlexNet}(\hat{x}_{t+n})\|_2^2$ is ℓ-2 distance between ground-truth and predicted in the feature space defined by the last convolution layer of a pre-trained AlexNet, and $\mathcal{L}_{adv} = -\log D([p_{t+n}, \hat{x}_{t+n}])$ is the adversarial loss. Finally, λ_{img}, λ_{feat}, and λ_{adv} are hyperparameters denoting how much each loss function will contribute to our overall loss function.

Discriminator: We input pose structure and video pairs to the discriminator, which outputs 1 if pose and video are real and from the same time slice and outputs 0 otherwise. The network architecture of the discriminator is also VGG-16. Our discriminator loss is defined as follows:

$$\begin{aligned}\mathcal{L} = &-\log D([p_{t+n}, x_{t+n}]) \\ &- 0.5\log(1 - D([p_{t+n}, \hat{x}_{t+n}])) \\ &- 0.5\log(1 - D([p_{t+n}, x_t]))\end{aligned}$$

Note that the first term incentivizes the discriminator to predict "real" given a real image and pose pair, the second term incentivizes the discriminator the predict "fake" given a real pose and fake image, and the third term incentivizes the discriminator to predict "fake" given an image at time t and a pose at a different time (so the discriminator can become sensitive to pose differences).

4. Experiments

Our experiments largely focus on human pose predictions and creating image analogies.

Table 1. Information of Penn Action Dataset

Actions		Annotated Joints	
1	baseball pitch	1	head
2	clean and jerk	2	left shoulder
3	pull ups	3	right shoulder
4	strumming guitar	4	left elbow
5	baseball swing	5	right elbow
6	golf swing	6	left wrist
7	push up	7	right wrist
8	tennis forehand	8	left hip
9	bunch press	9	right hip
10	jumping jacks	10	left knee
11	sit ups	11	right knee
12	tennis serve	12	left ankle
13	bowling	13	right ankle
14	jump rope		
15	squats		

4.1. Dataset

We use the Penn Action Dataset [12] for training and testing. The dataset consists of 2325 videos of human actions. The 15 actions in the dataset are baseball pitches, baseball swings, bench press, and some others. Each image has a 13 annotated joint locations stored as xy coordinates denoting joint coordinate position in image. The joints include the head, left shoulder, right shoulder, left elbow, right elbow, and some others. Most videos are 480x270 and consist around 50-150 frames. The details of the dataset is shown in Table 1.

4.2. Pose Estimation

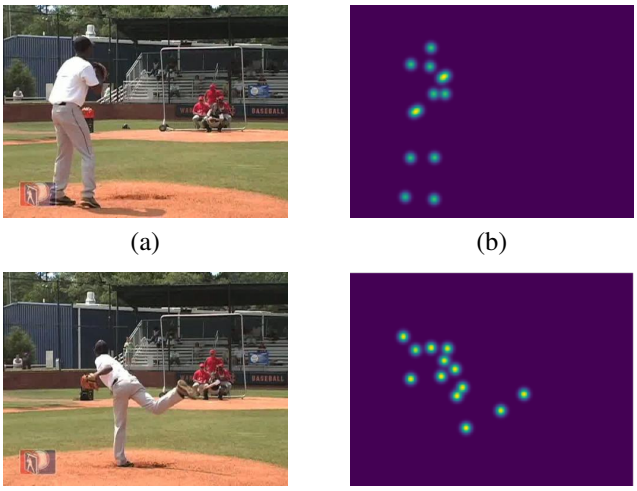
Using the annotated joint locations from the Penn Action Dataset, we create a corresponding heatmap by drawing a circle of radius 5 at each location and applying a Gaussian blur with $\sigma = 5$ (illustrated in Fig. 4). This heatmap will represent our pose estimation input into the image analogy network.

4.3. Pose Prediction

We apply a 2-layer neural network function to the output of the LSTM architecture described above. The network consists of a 1024-node hidden layer followed by a 100-node hidden layer, both with sigmoid activation function. We use $k = 15$ input images to encode our LSTM network and decode $T = 45$ prediction images. We trained the network on 73 squats video and ran tests on 5 videos.

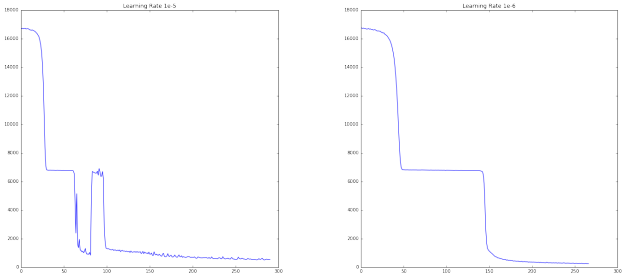
4.4. Image Analogy

For the image analogy network, we trained the network on squats videos. We converted all videos to grayscale and resize to 224x224. Our network parameters are $\lambda_{img} = 1$, $\lambda_{feat} = 1$ and $\lambda_{adv} = 0.1$. We define each training epoch



(a) (b) (c) (d)

Figure 4. Creating heatmaps from annotated labels from Penn Dataset



(a) (b)

Figure 5. (a) Learning rate 1e-5. (b) Learning rate 1e-6. In the images above, the final loss for 1e-5 is 600 and the final loss for 1e-6 is 200. Notice 1e-6 is much smoother and more stable. It was able to converge to a better loss value of 200 in a shorter amount of time and does not have the problem of jumping in and out of optima.

to be 1000 image analogy pairs and train on 200 epochs. The learning rate for the generator network and the discriminator network is $1e^{-6}$.

We did the majority of the training on a GTX 1070 and the above parameters took around 12 hours to train. We used a stochastic gradient descent optimizer with a mini-batch size of 4. Because the network is so large (several VGGs), 4 was the maximum mini-batch size we could work with.

5. Results and Analysis

5.1. Results on Pose Prediction

Our results show that the LSTM network is able to capture the movement of squats accurately. There is high variation in the form and speed of how people do squats. The LSTM model is able to capture an "average" prediction model based on the trained videos. Thus, though the predicted joint locations do not align perfectly in our tests, the predictions show a convincing alternative to the ground truth, as shown in Fig. 6 and 7.

5.2. Analysis of Pose Prediction

Our initial test with a simple sequence-to-sequence network with 128 hidden units and no fully connect output layer did not work well in predicting poses for test videos. The simple sequence-to-sequence network was able to overfit to training data, but was unable to make meaningful predictions for test images. By increasing the number of hidden units in the LSTM block and adding two fully connected layers to the output of the LSTM, we were able to create a working prediction model.

5.3. Results on Image Analogy

Our image analogy network partially captures the motion of the human. From the results in Fig. 8, we can see that the network easily converged on a mean image prediction of the training images and only partially used the pose information to generate accurate predictions. We tried contacting the authors of the original paper for insight into how they trained their network and what hyperparameters they used, but got no response. Thus, in an attempt to get better performance, we chose to tune hyperparameters (details shown in Section 5.4). However, hyperparameter tuning does not led to very significant improvement in results.

5.4. Analysis of Image Analogy

In our initial attempt to create the image analogy network, we had a bug that scrambled the heatmaps (i.e. inputted heatmaps did not correspond to the correct frame). The result is that the image outputted is a mean image of the training frames for each video. Interestingly, after fixing this heatmap correspondences bug, the network still produces the same result.

Our first instinct was that our hyperparameters λ_{img}, λ_{feat} and λ_{adv} were not tuned properly. We did cross validation across 9 different combinations of parameters, all producing similar or worse results. The values we tried for [λ_{img}, λ_{feat}, λ_{adv}] were [100,50,1], [50,50,1], [10,50,1], [1,50,1], [1,1,0.1], [0.5,5,5], [0.5,10,10], [0.5, 25, 25] and [0.01,1,1]. We noticed that the l2 loss of the images contributed the most to the loss and to the overall quality of the image. With multiple VGG networks (totaling 4GB in weights), running numerous tests to tune hyperparameters is very time consuming. Hence, we stopped after 9 variations.

The second observation is the information from the pose seems to be ignored. We believe putting more emphasis on

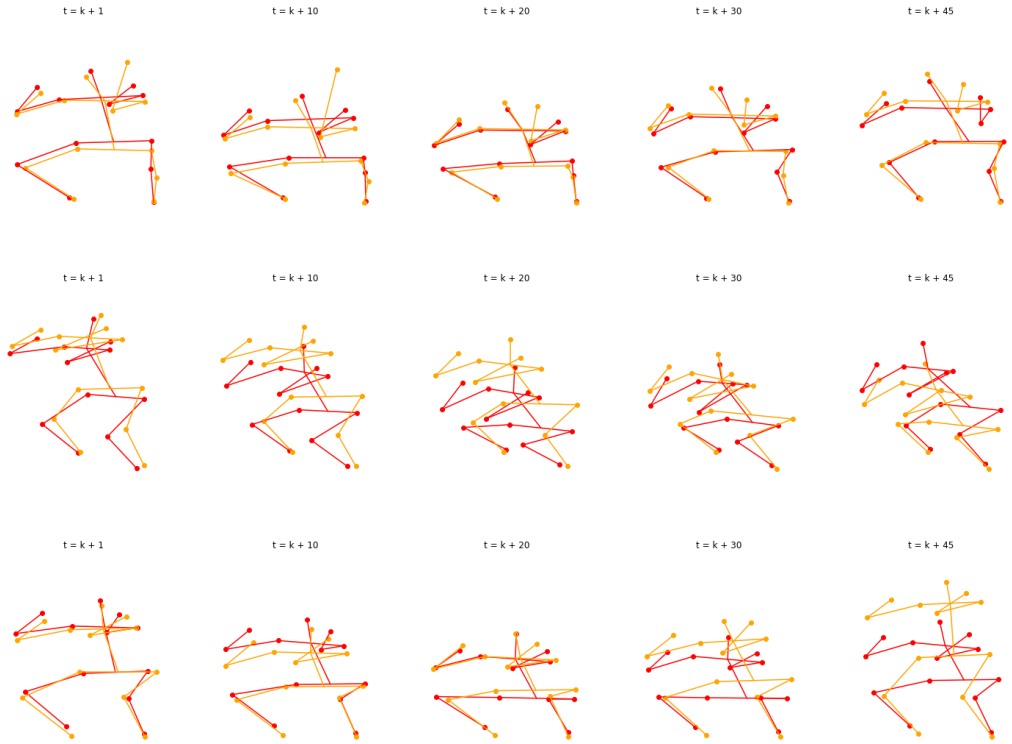


Figure 6. LSTM prediction on fully unoccluded set of joints. In our model k = 15 is the number of input frames. The predictions (**red**) do not fully match the ground truth (**orange**) but captures the motion quite well.

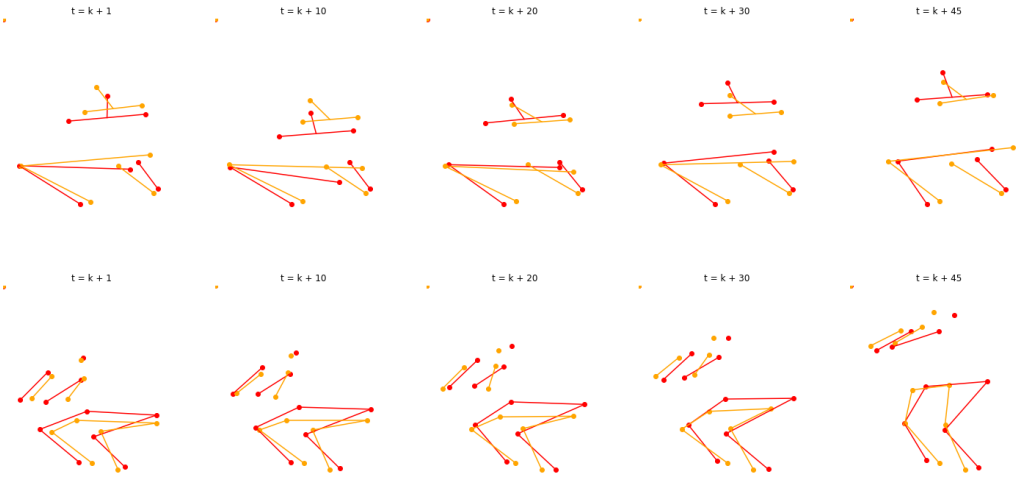


Figure 7. LSTM prediction on cases where some joints are occluded, in which case (x, y) = (0, 0) for those joints. The rest of the joints are predicted well.

the pose in the latent space may help fix our issue. This is something we have not tried and can be considered for future work.

One notable part is that there is no released source code for the architectures proposed and we wrote all the code from scratch in Tensorflow.

Finally, finding an optimal learning rate proved to be very crucial for proper convergence. Because our network is very large, we were unable to load optimizers like Adam, Momentum, or Adagrad in Tensorflow. These optimization methods store additional variables for each weight, which exceed our GPU vram limit of 8GB. We resorted to stan-

dard GradientDescentOptimizer with a very small batch size, which requires a more precisely defined learning rate. In Fig. 5. we show the effect of our training loss on two learning rates: $\epsilon = 1e^{-5}$ and $\epsilon = 1e^{-6}$. We can see that $\epsilon = 1e^{-5}$ is a suboptimal learning rate that is too high and was not able to converge as quickly as $\epsilon = 1e^{-6}$. Notice the instability associated with $\epsilon = 1e^{-5}$ loss. Training the network can take many hours, making it an important task to find a good learning rate to speed up training and converge to an optimal value.

6. Conclusion and Future Work

In this paper, we present a hierarchical model of pixel-level video prediction. Using human action video dataset as benchmark, we demonstrate our hierarchical prediction approach that our hierarchical model is able to predict pose heatmaps.

The success of our work is that the model can predict long-term pose structure from video frames. This is significant because it supports the success of hierarchical model for video frame prediction. The limitation of our work is that the analogy network does not yield very realistic result.

For future work, we will work on making long-term RGB video frame prediction. This work only generates a single feature trajectory and work can be done on generating predictions for multiple features. Finally, we can work on generalizing pose prediction to segmentation predictions to work for arbitrary objects (and not just human poses).

References

[1] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016.

[2] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016.

[3] I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[4] R. Goroshin, M. F. Mathieu, and Y. LeCun. Learning to linearize under uncertainty. In *Advances in Neural Information Processing Systems*, pages 1234–1242, 2015.

[5] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[6] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[7] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh. Action-conditional video prediction using deep networks in atari games. In *Advances in Neural Information Processing Systems*, pages 2863–2871, 2015.

[8] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee. Deep visual analogy-making. In *Advances in Neural Information Processing Systems*, pages 1252–1260, 2015.

[9] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*, pages 843–852, 2015.

[10] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. *arXiv preprint arXiv:1704.05831*, 2017.

[11] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[12] W. Zhang, M. Zhu, and K. G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2248–2255, 2013.

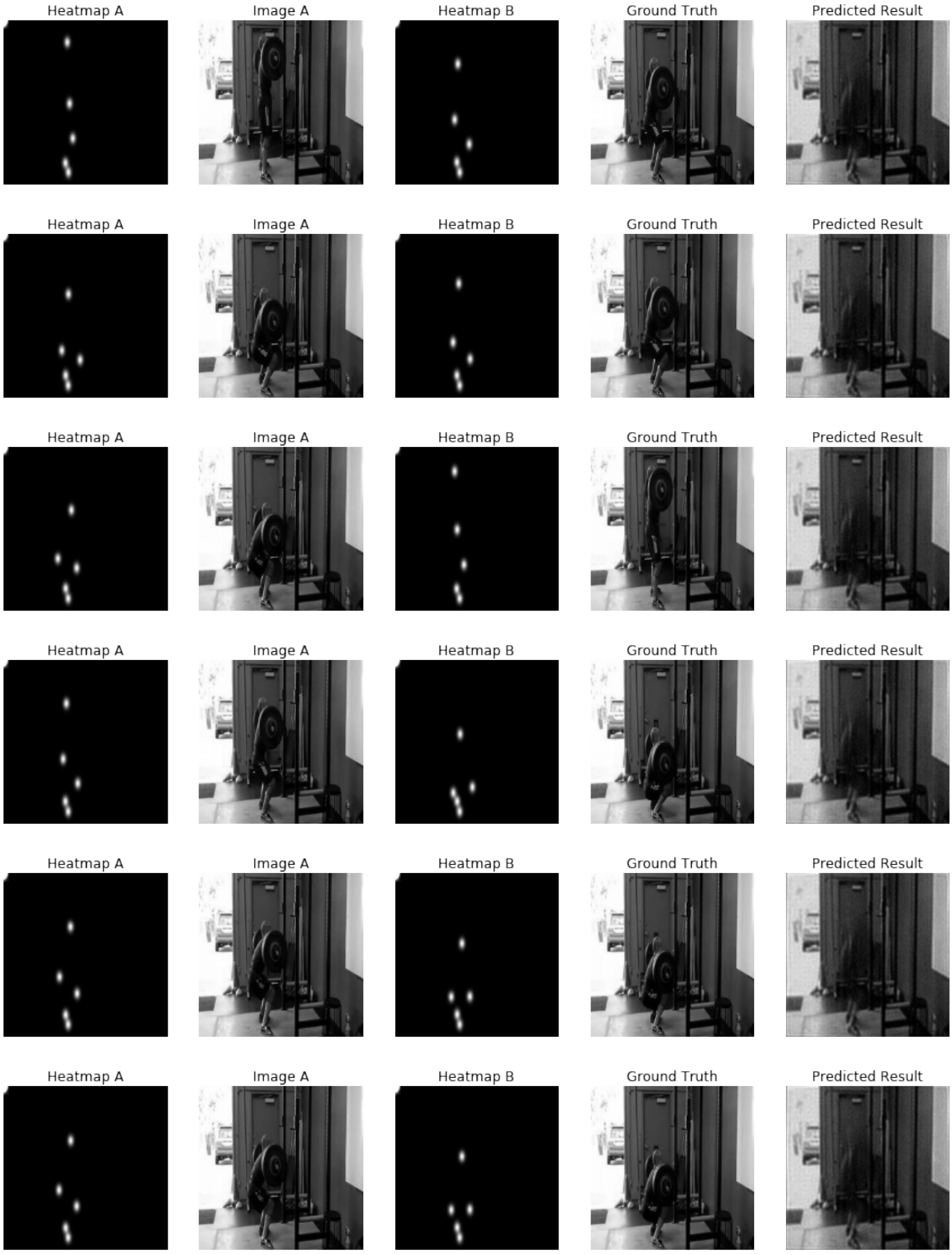


Figure 8. The results of image analogy network. The prediction partially captures human motions.