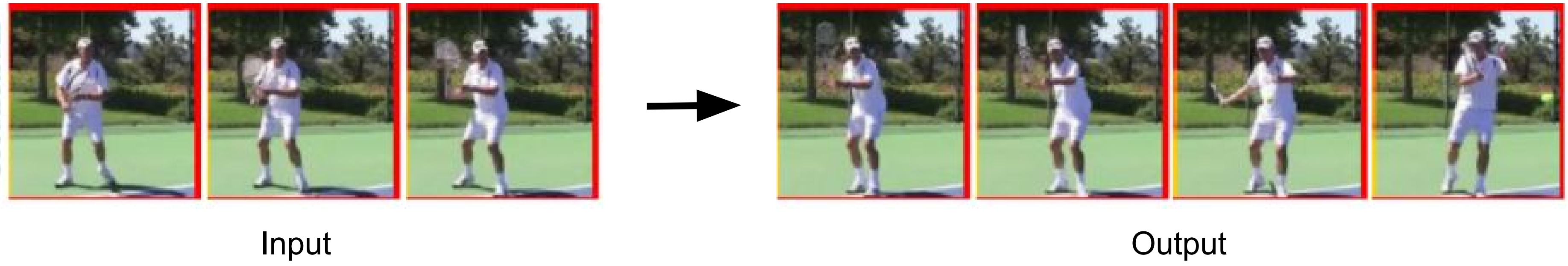
Hierarchical Model for Long-term Video Prediction

Peter (Jiayun) Wang, Zhongxia Yan, Jeffrey Zhang

Problem Statement

- Given the previous frames of the video as input, we want to get the long-term frame prediction.
- Initial goal was to predict the motion segmentation of a video and use segmentation to predict full video
- Adopted human pose prediction method based on: Villegas, Ruben, et al. "Learning to Generate Long-term Future via Hierarchical Prediction." arXiv:1704.05831 (2017)

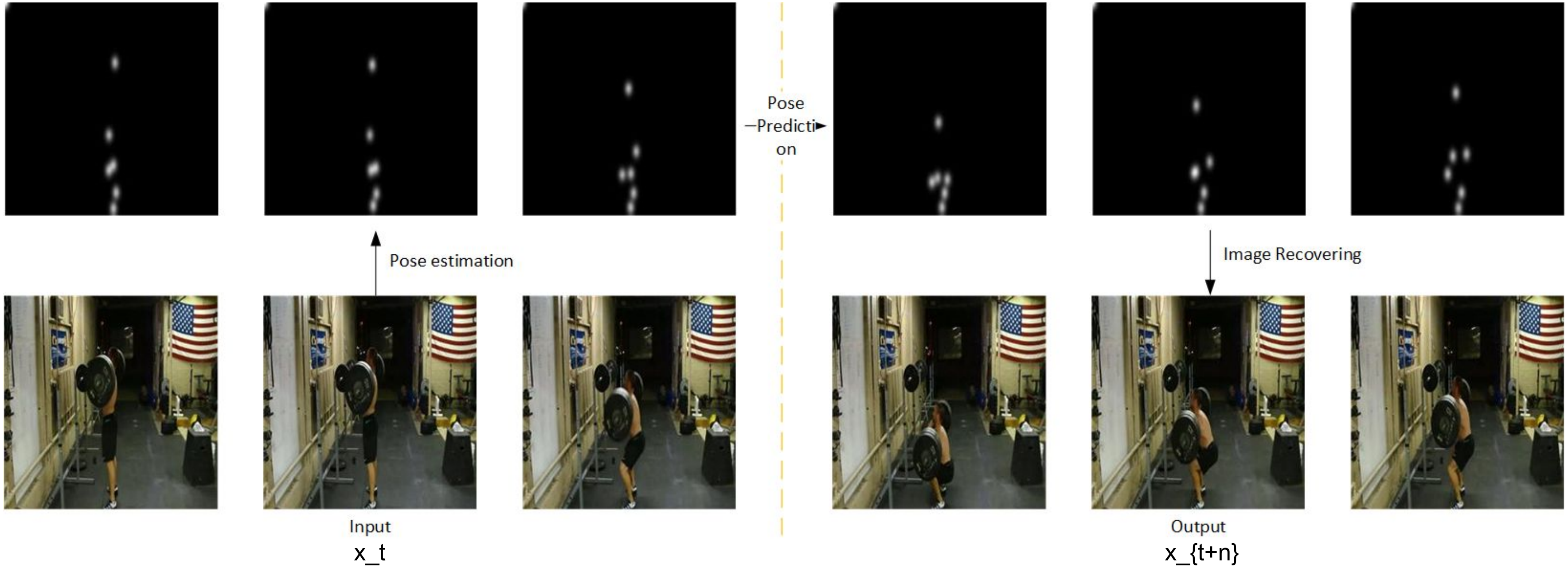


Related Work

- Video Prediction Based on Deep Voxel Flow (Liu et al., 2017)
- Recursive Approach to Pixel-wise Video Prediction (Oh et al., 2015)
- Deep Multi-scale Video Prediction Beyond Mean Square Error (Mathieu et al., 2015)
- Generating Videos with Scene Dynamics (Vondrick et al, 2016)

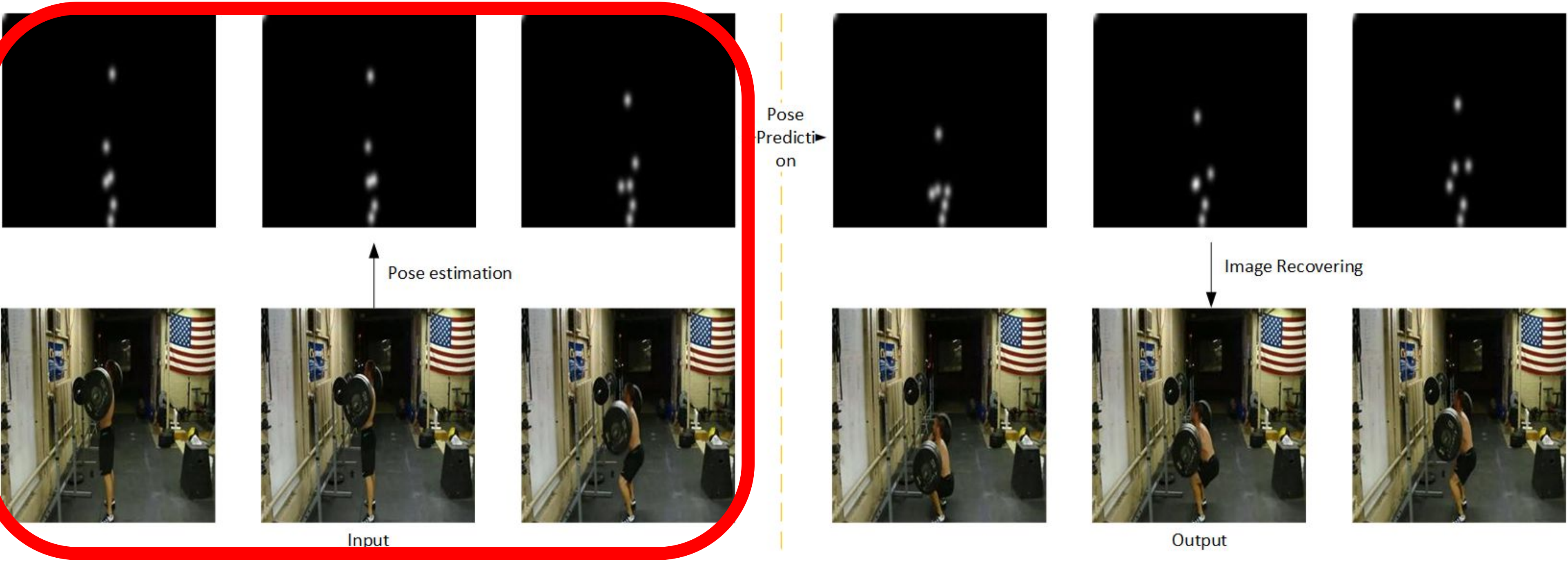
Method

1. Pose Estimation
2. Pose Prediction
3. Image Recovery



1. Pose Estimation

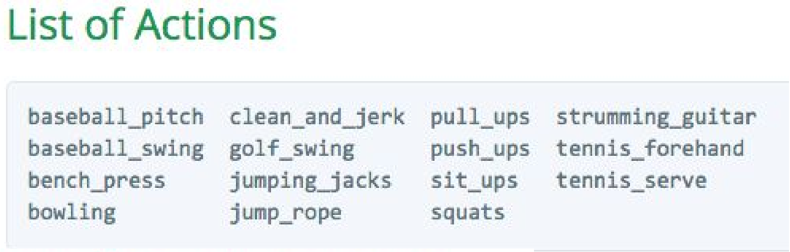
- For input frame, we generate the corresponding heatmaps



Penn Action Dataset

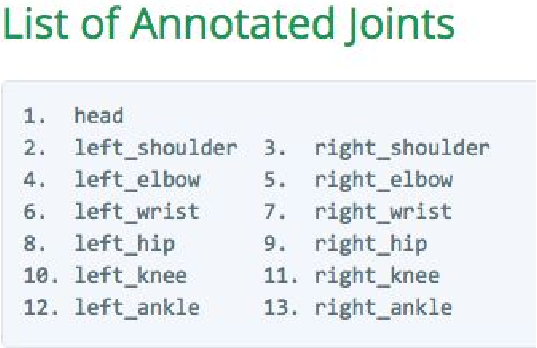
http://dreamdragon.github.io/PennAction/

- 2325 videos (50-150 frames each)



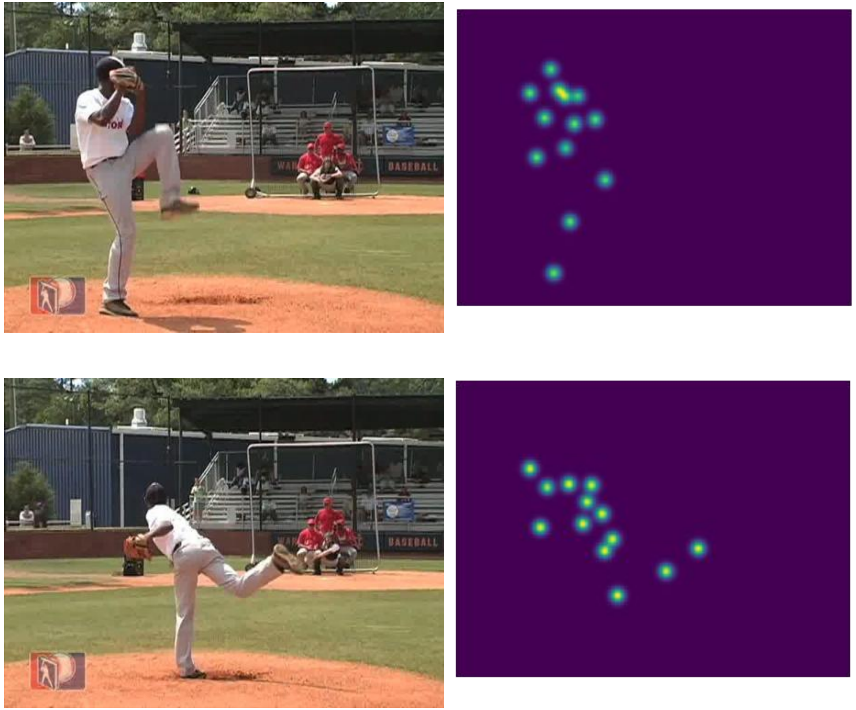
List of Actions

```
baseball_pitch   clean_and_jerk   pull_ups   strumming_guitar
baseball_swing   golf_swing       push_ups   tennis_forehand
bench_press      jumping_jacks    sit_ups    tennis_serve
bowling          jump_rope        squats
```



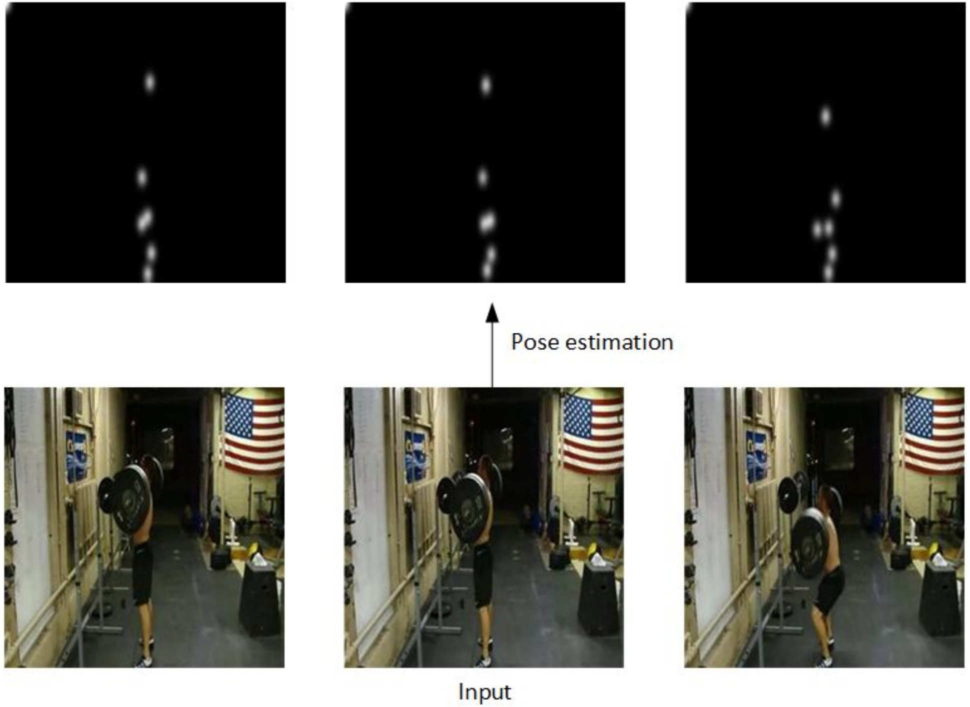
List of Annotated Joints

```
1.  head
2.  left_shoulder  3.  right_shoulder
4.  left_elbow     5.  right_elbow
6.  left_wrist     7.  right_wrist
8.  left_hip       9.  right_hip
10. left_knee      11. right_knee
12. left_ankle     13. right_ankle
```



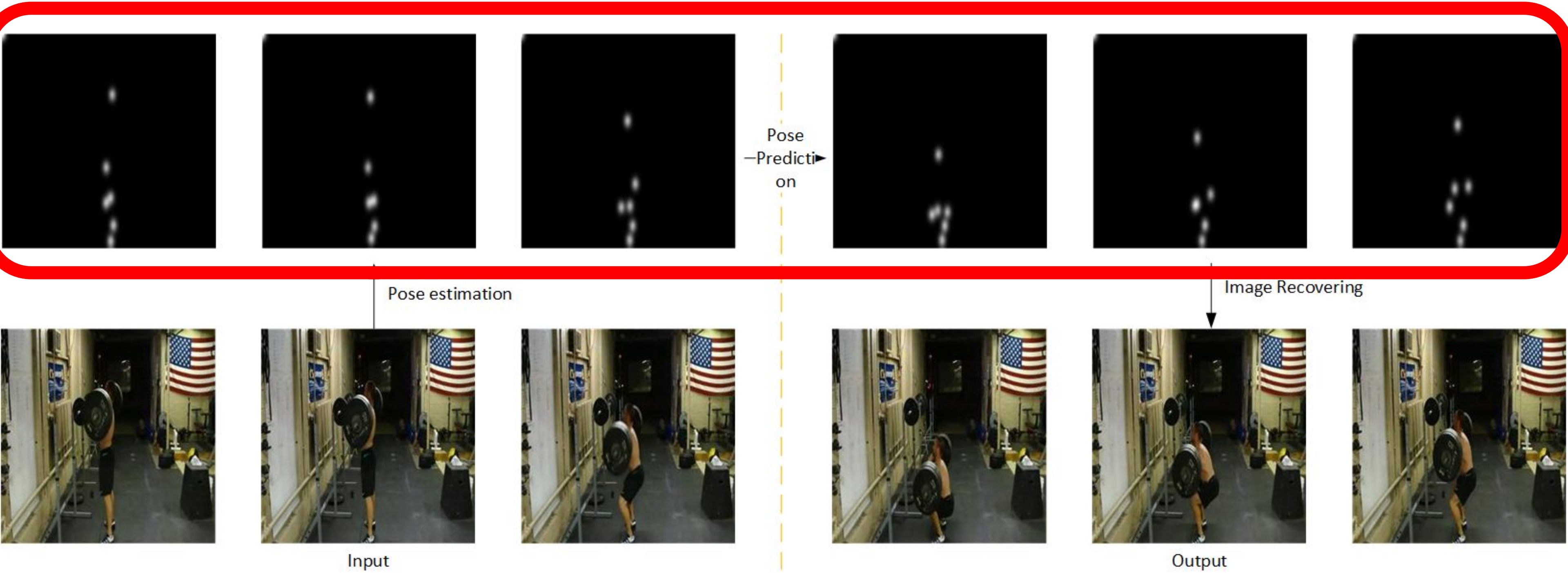
1. Pose Estimation

- Penn Action dataset already have annotation of the pose
- For general videos, Hourglass network (Newell et al., 2016) is used to generate the pose heatmap. (Future work)



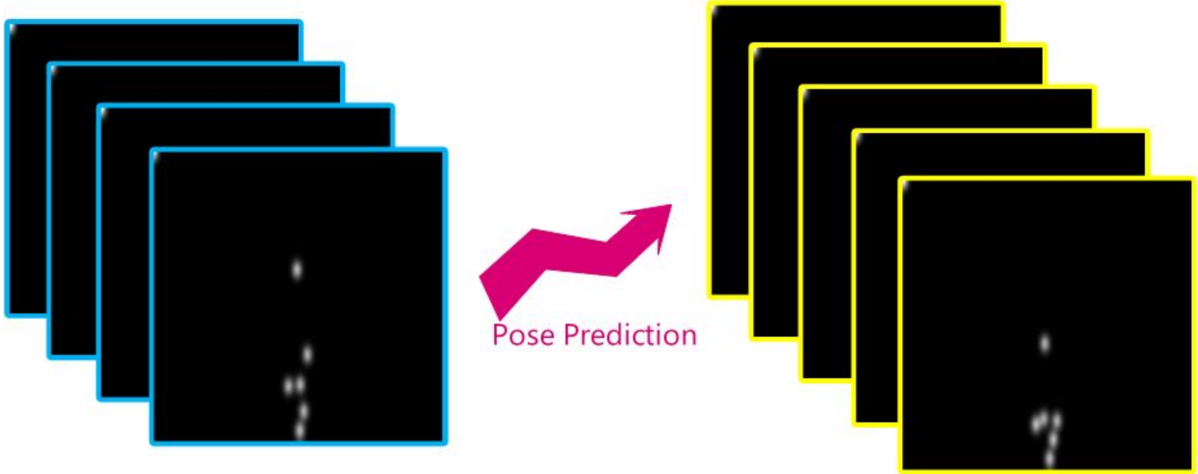
2. Pose Prediction

- We use **Pose Prediction Network** to predict the long-term future pose.



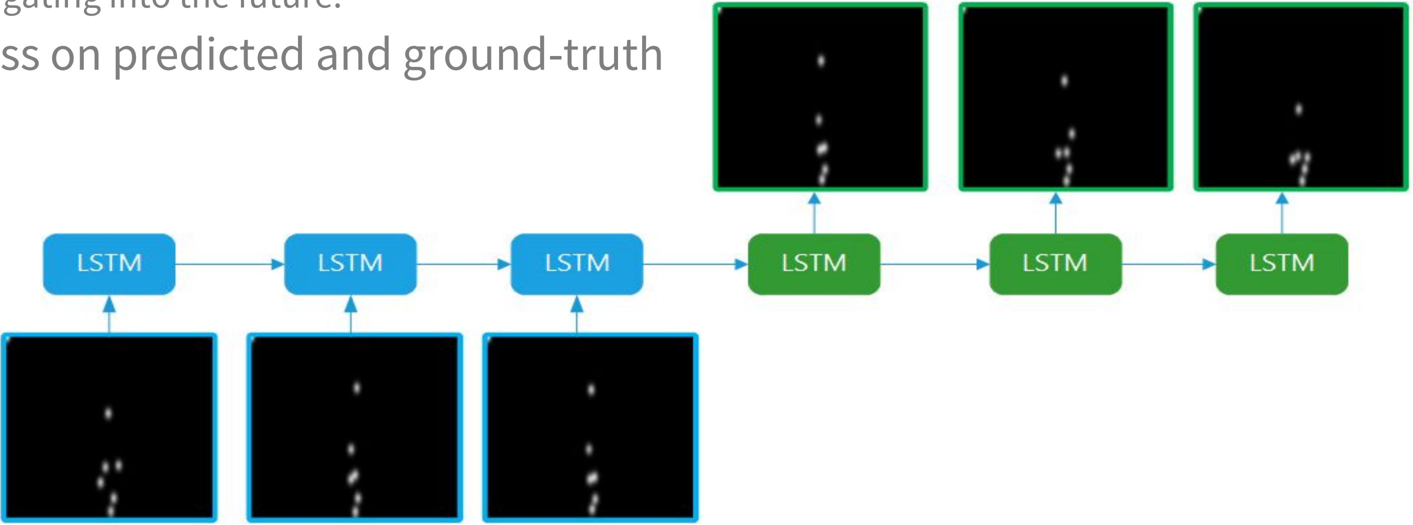
2. Pose Prediction

- We use LSTM to predict the long-term pose joint locations

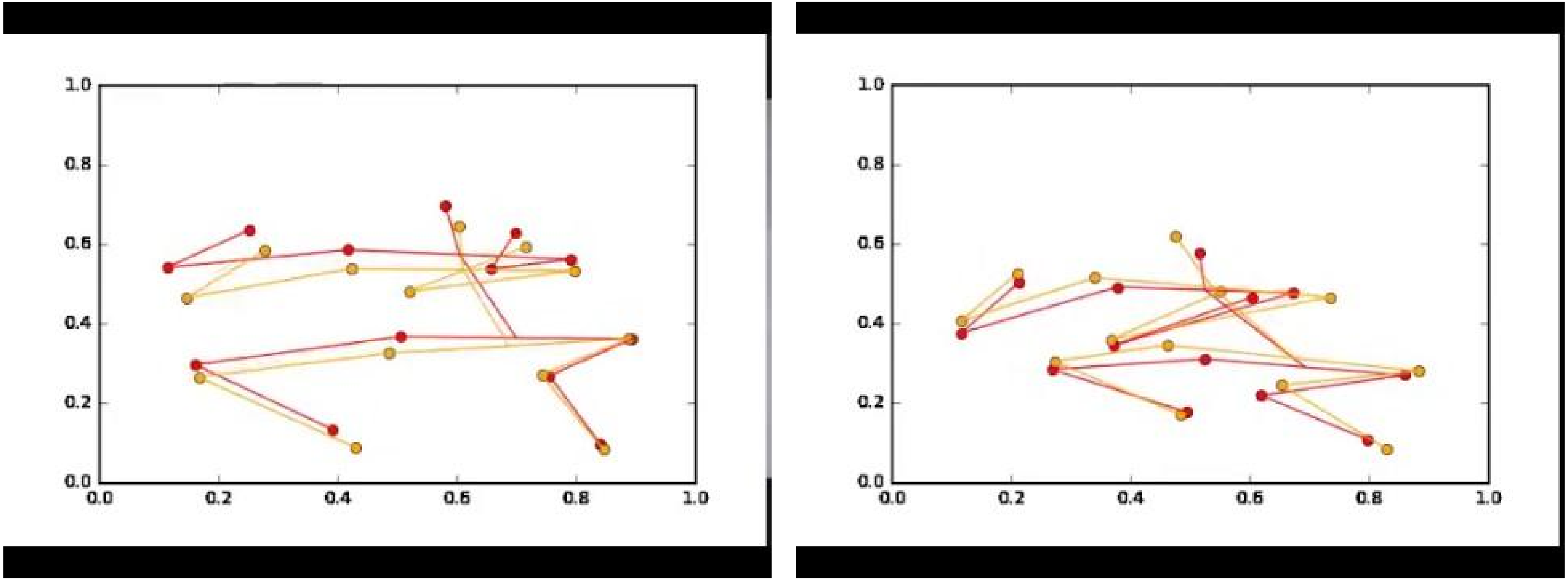


2. Pose Prediction

- We encode the network with k = 15 input frames
- We decode the next T = 45 frames using the network
  - During prediction, we only feed in 0
  - We do not feed in the output from the previous frame to prevent errors from propagating into the future.
- Loss: l_2 loss on predicted and ground-truth



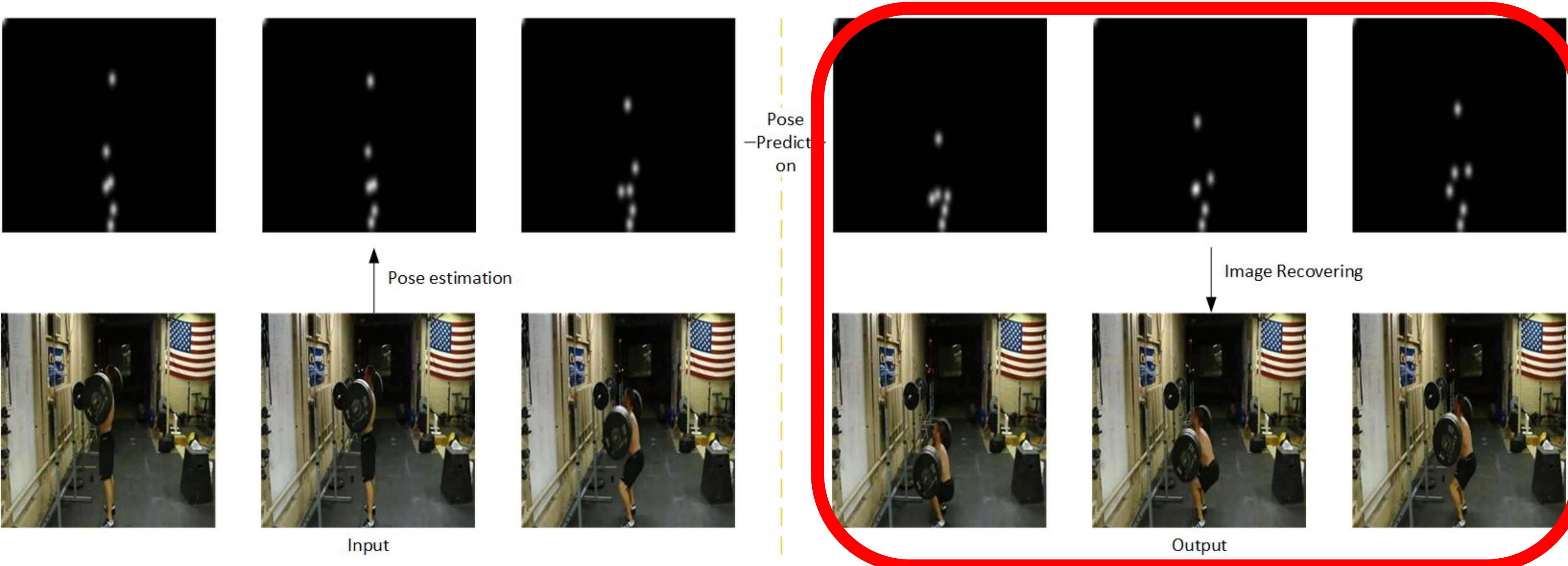
LSTM pose prediction on training set (red: prediction, orange: ground truth)



LSTM pose prediction on test set (red: prediction, orange: ground truth)

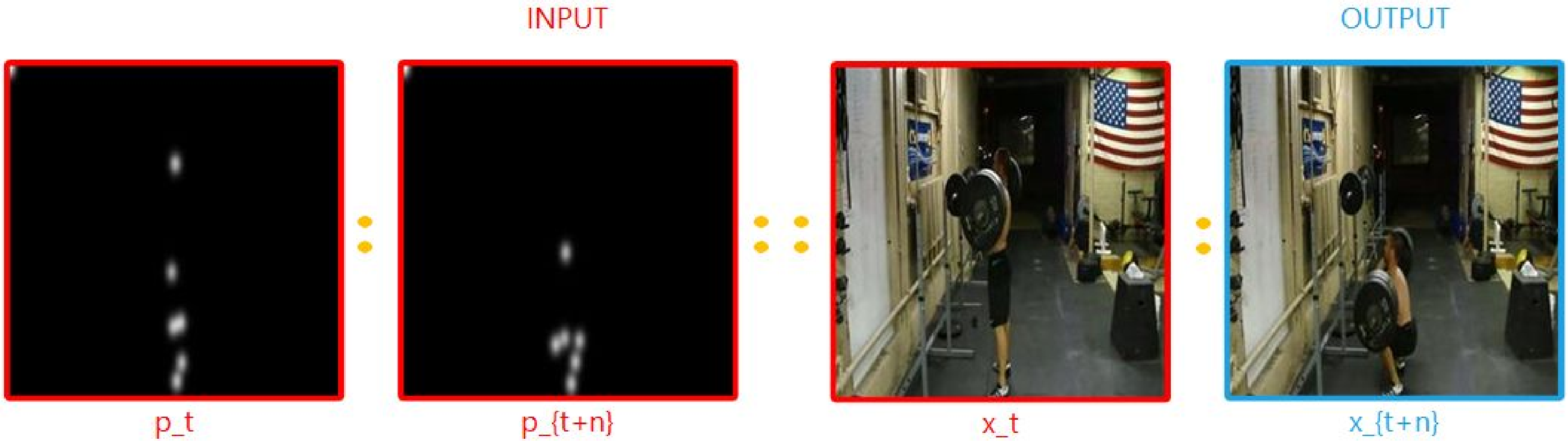
3. Image Recovery (Image Analogy Network)

- We use **Image Analogy Network** to recover the real image prediction from pose heat map.



3. Image Recovery (Image Analogy Network)

- We use **Image Analogy Network** to recover the real image prediction from pose heat map.

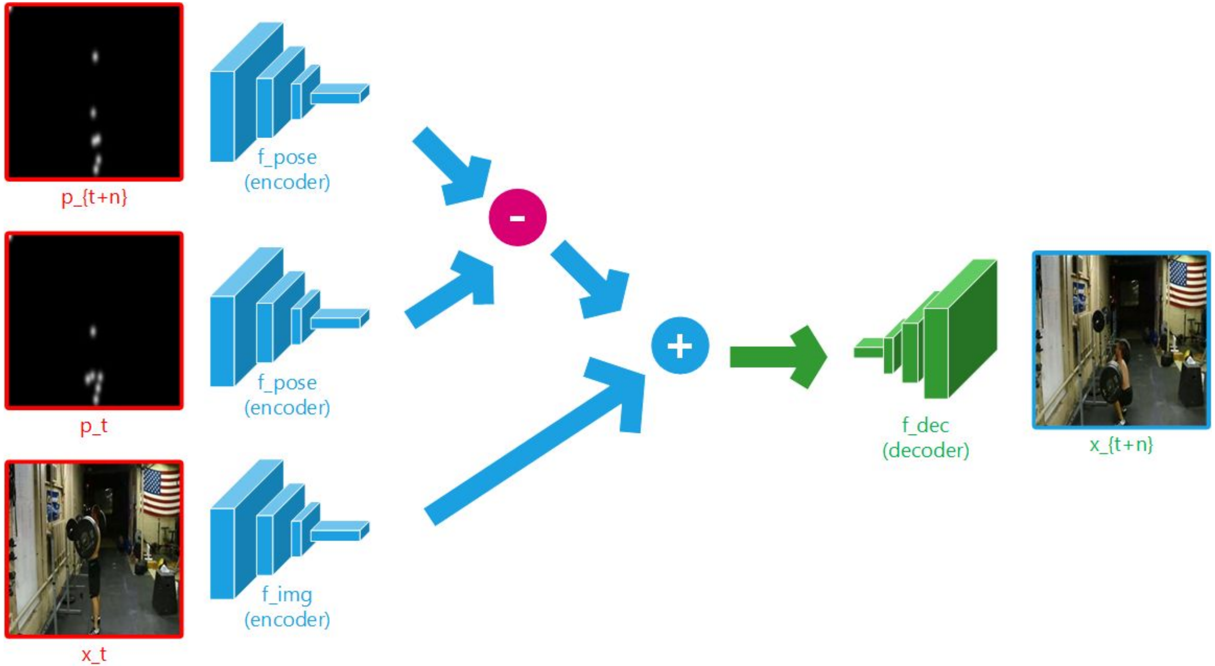


(predicted by LSTM)

3. Image Recovery (Image Analogy Network)

Generator Architecture:

- Inputs:
    - poses at time t and t + n
    - video frame at time t
- Output:
    - video frame at time t + n
- Encoders are VGG-based
- Decoder is a deconv VGG



$$\hat{\mathbf{x}}_{t+n} = f_{\text{dec}}\left(f_{\text{pose}}\left(g\left(\hat{\mathbf{p}}_{t+n}\right)\right) - f_{\text{pose}}\left(g\left(\mathbf{p}_t\right)\right) + f_{\text{img}}\left(\mathbf{x}_t\right)\right)$$

3. Image Recovery (Image Analogy Network)

Generator Loss Function:

$$\mathcal{L} = \mathcal{L}_{\text{img}} + \mathcal{L}_{\text{feat}} + \mathcal{L}_{\text{Gen}}$$

L_{img} is the loss in image space, L_{feat} is loss in AlexNet and VGG feature spaces, L_{Gen} is adversarial loss

$$\mathcal{L}_{\text{img}} = \|\mathbf{x}_{t+n} - \hat{\mathbf{x}}_{t+n}\|_2^2$$

$$\mathcal{L}_{\text{feat}} = \|C_1(\mathbf{x}_{t+n}) - C_1(\hat{\mathbf{x}}_{t+n})\|_2^2 + \|C_2(\mathbf{x}_{t+n}) - C_2(\hat{\mathbf{x}}_{t+n})\|_2^2$$

$$\mathcal{L}_{\text{Gen}} = -\log D([\mathbf{p}_{t+n}, \hat{\mathbf{x}}_{t+n}])$$

3. Image Recovery (Image Analogy Network)

Discriminator Architecture:

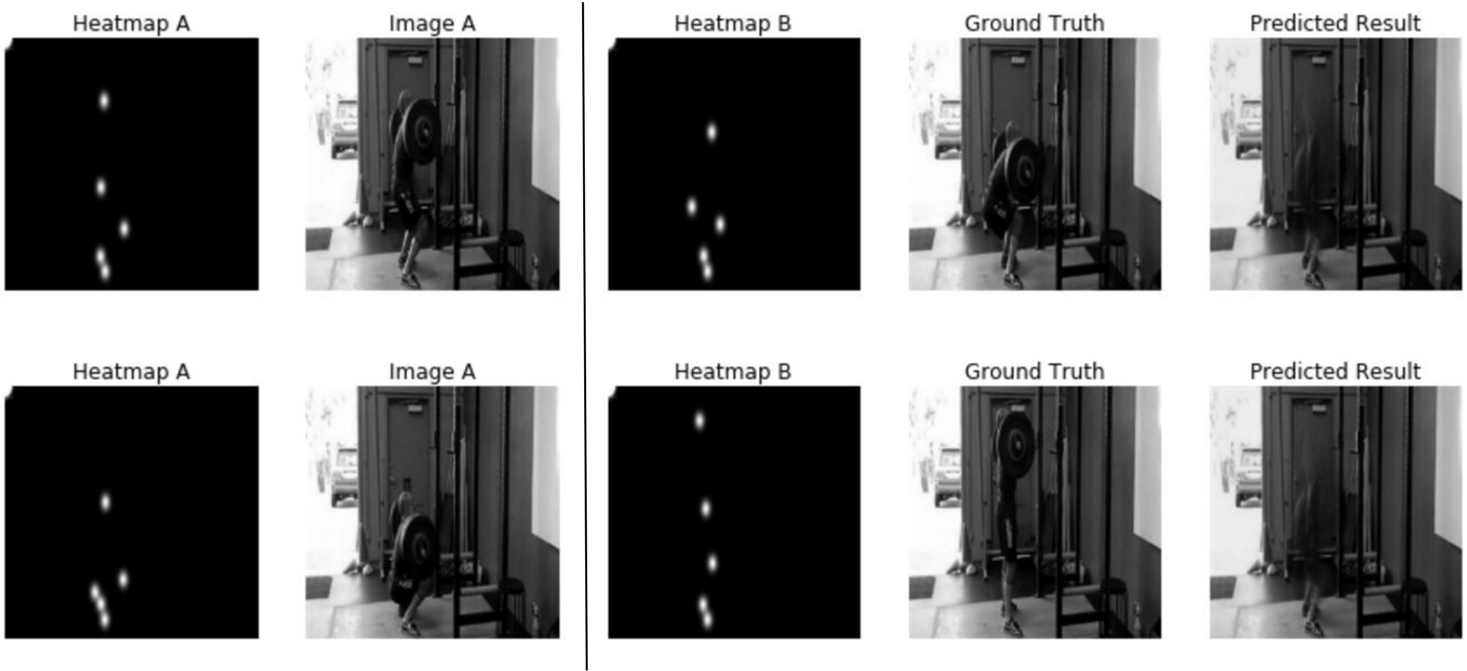
- Input: pose and video pairs
- Output: 1 if pose and video are real *and* from the same time slice
- VGGs for pose and video

Discriminator Loss Function:

$$\mathcal{L}_{Disc} = -\log D([p_{t+n}, x_{t+n}]) - 0.5\log(1 - D([p_{t+n}, \bar{x}_{t+n}])) - 0.5\log(1 - D([p_{t+n}, x_t]))$$

3. Image Recovery (Image Analogy Network)

Results:



x_t

x_{t+n}

Future Work

1. Improving the results on image analogy network
2. Train image analogy network also on RGB frames
3. Use hourglass network to generate pose heatmaps
4. Use Caroline’s pose transfer instead?
5. Explore replacing pose estimation with motion segmentations

Questions?

Outline

- 45 seconds : background, inspiration, progression. (Basically our thought process:
    - Yann LeCunn’s paper
    - predicting segmentation movement (reflects human prediction of general features rather than pixel by pixel)
        - Idea: train segmentation predictions then fill in high frequency data
- 2 minutes 30 seconds: Our initial approach + detail Villegas paper + our implementation
    - Trained model to predict joint movements (really still pretty blurry)
    - Found Villegas paper using LSTM to predict joint locations and using image analogies to fill in data
- 45 seconds: Results