Sequence-to-Sequence Learning for Human Pose Correction in Videos

Sirnam Swetha
CVIT, KCIS
IIIT-Hyderabad
Hyderabad, India
sirnam.swetha@research.iiit.ac.in

Vineeth N Balasubramanian
Dept. of CSE
IIT-Hyderabad
Hyderabad, India
vineethnb@iith.ac.in

C.V. Jawahar
CVIT, KCIS
IIIT-Hyderabad
Hyderabad, India
jawahar@iiit.ac.in

***Abstract*—The power of ConvNets has been demonstrated in a wide variety of vision tasks including pose estimation. But they often produce absurdly erroneous predictions in videos due to unusual poses, challenging illumination, blur, self-occlusions etc. These erroneous predictions can be refined by leveraging previous and future predictions as the temporal smoothness constrain in the videos. In this paper, we present a generic approach for pose correction in videos using sequence learning that makes minimal assumptions on the sequence structure. The proposed model is generic, fast and surpasses the state-of-the-art on benchmark datasets. We use a generic pose estimator for initial pose estimates, which are further refined using our method. The proposed architecture uses Long Short-Term Memory (LSTM) encoder-decoder model to encode the temporal context and refine the estimations. We show 3.7% gain over the baseline Yang & Ramanan (YR) [1] and 2.07% gain over Spatial Fusion Network (SFN) [2] on a new challenging YouTube Pose Subset dataset [3].**

***Keywords*-Pose estimation; sequence to sequence learning; LSTM;**

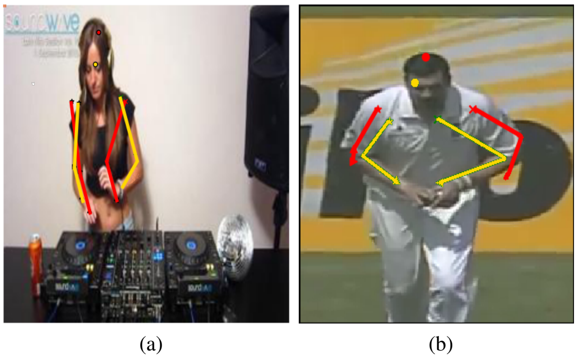


Figure 1: **Human Pose Correction.** Pose predictions on sample images from the datasets used in this work. Red and yellow correspond to joints predicted using baseline (initialization) and our method respectively. (Best viewed in color)

I. Introduction

Estimating 2D human pose from images is a challenging task with many applications in computer vision, such as motion capture, sign language, human-computer interaction and activity recognition. Profuse amount of work has been done on articulated pose estimation from single images [4], [5], [6], [7], [8], [9]. Despite steady advances, pose estimation remains as an intricate problem. Recent advances in 2D human pose estimation exploit complex appearance models and more recently convolutional neural networks (ConvNets) [10], [11], [12], [2], [13], [14], [15]. We focus on the task of 2D human pose estimation in videos in the wild : single-view, uncontrolled settings typical in movies, television and amateur videos. This task is made difficult by the considerable background clutter, camera movement, motion blur, poor contrast, body pose and shape variation, as well as illumination, clothing and appearance diversity. Even the state-of-the-art ConvNets often produce erroneous predictions in videos due to these challenges (Figure 1).

To date, CNN models for video processing have successfully considered learning of 3-D spatio-temporal filters over raw sequence data [16], and learning of frame-to-frame representations which incorporate instantaneous optic flow or trajectory-based models aggregated over fixed windows or video shot segments [17]. Such models explore two extrema of perceptual time-series representation learning: either learn a fully general time-varying weighting, or apply simple temporal pooling. Following the same inspiration, the video sequence learning models which are also deep over temporal dimensions; i.e., have temporal recurrence of latent variables. Recurrent Neural Network (RNN) models are deep in time - explicitly so when unrolled - and form implicit compositional representations in the time domain.

Pose predictions from neighbouring frames are not independent of each other and they form a sequence. We formulate the correction problem as sequence-to-sequence learning problem, while leveraging the temporal smoothness implicitly encoded in the target sequence. There have been a number of related attempts [18], [19], [20], [21] to address the general sequence-to-sequence learning problem with neural networks. Instead of correction of one prediction at a time (CRF based post processing), this model can capture the complex pose configurations over time while the body undergoes numerous appearance changes, resulting in a more reliable correction model.

In this paper, we propose pose correction model in videos as a sequence-to-sequence learning problem (Figure 2). The neural network architecture, which we will refer to as an LSTM encoder-decoder, consists of two recurrent neural networks that act as an encoder and a

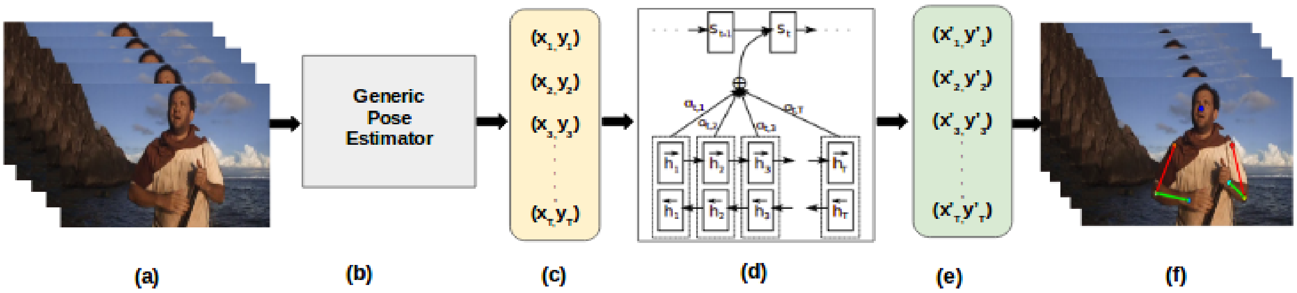


Figure 2: **Overview of our method.** (a) Input videos, (b) Generic pose estimator, (c) Initial pose estimates (x_i, y_i) for all joints, (d) Correction model where h_t, s_t are the hidden states at time t and a_{ij} is an alignment model which scores how well the inputs around position j and the output at position i match, (e) Refined pose estimates (x'_i, y'_i) for all joints and (f) Pose visualization. A bidirectional LSTM encoder is used in the refinement model as shown in (d). The correction model corrects the erroneous poses (predicted by a generic pose estimator (b)).

decoder pair. The encoder maps an input source sequence to a fixed-length vector, and the decoder maps the vector representation back to a target sequence. The two networks are trained jointly to maximize the conditional probability of the target sequence given a source sequence. In Section 2, we present related work and also background for the architecture followed by discussion on our approach in Section 3. We show the results of proposed approach in Section 4 followed by conclusions and pointers to future work in Section 5.

II. Related Work

Early methods using ConvNets regressed the pose coordinates of human joints directly (as (x, y) coordinates) [14]. An alternative, which improved over earlier ConvNets, is an indirect prediction by first regressing a heatmap over the image for each joint, and then obtaining the joint position as a mode in this heatmap [11], [2], [15].

There were attempts to use motion cues in videos such as optical flow to handle the erroneous predictions and also to prune the poses which are not possible kinematically [2]. But optical flow itself is erroneous (also mentioned in [2]). For videos with high variations across neighborhood frames, the optical flow errors commensurates to the motion across frames. Charles *et al.* [3] use person-specific information to localize joints and boosts the predictions along with occlusion-aware methods. The proposed methodology can be used along with any existing pose estimation method as a correction mechanism.

We propose a simple and generic framework for error refinement of joint predictions using an encoder-decoder attention model for sequence learning. A bidirectional encoder is used by default. There is no hidden state transfer in this model. It is a generic, reliable model which captures the latent variables using non-linear mechanism. Graves *et al.* [20] introduced a novel differentiable attention mechanism that allows neural networks to focus on different parts of input, and an elegant variant of this idea was successfully applied to machine translation by Bahdanau *et al.* [21]. Our approach is very close to [21], which learns a soft alignment between the input and output sequences which improves the performance (only for text, however). The Connectionist Sequence Classification is another popular technique for mapping sequences to sequences with neural networks, although it assumes a monotonic alignment between the inputs and the outputs [22].

RNN Encoder-Decoder. In the Encoder-Decoder framework (proposed by Cho *et al.* [19] and Sutskever *et al.* [23]), an encoder reads the input sequence, a sequence of vectors $x = (x_1, \cdots, x_{T_x})$, into a vector c. The most common approach is to use an RNN such that $h_t = f(x_t, h_{t-1})$ and $c = q(\{h_1, \cdots, h_{T_x}\})$ where $h_t \in R^n$ is a hidden state at time t, and c is a vector generated from the sequence of the hidden states. f and q are some nonlinear functions. Sutskever *et al.* [23] used LSTM as f and $q(\{h_1, \cdots, h_T\}) = h_T$, for instance.

The decoder is trained to predict the next token $y_{t'}$ given the context vector c and all the previously predicted tokens $\{y_1, \cdots, y_{t'-1}\}$. The decoder defines a probability over the mapping y by decomposing the joint probability into the ordered conditionals:

$$p(y) = \textstyle\prod_{t=1}^{T} p(y_t | \{y_1, \cdots, y_{t-1}\}, c)$$

where $y = \{y_1, \cdots, y_{T_y}\}$. With an RNN, each conditional probability is modeled as

$$p(y_t | \{y_1, \cdots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c) \tag{1}$$

where g is a non-linear, potentially multi-layered, function that outputs the probability of y_t, and s_t is the hidden state of the RNN.

III. Pose Correction Model

An overview of our algorithm is shown in Figure 2. Our approach can be broadly divided into 2 stages. Each stage is independent, and the details of each stage are discussed below.

Initialization. Our method receives frames from videos and generates initial pose estimates for all the frames independently. We can use any generic pose estimator to generate initial pose estimates. It is often observed that these estimates are erroneous in videos, due to self-occlusion, blur, unusual poses, etc (Figures 1 and 3). For our experiments, we use the Spatial Fusion Network (SFN) [2] and the more traditional Yang & Ramanan [1] models to generate initial pose estimates.

A. General Decoder

The conditional probability in Eq. 1 is defined as

$$p(y_i|\{y_1, \cdots, y_{i1}\}, c) = g(y_{i-1}, s_i, c_i) \tag{2}$$

where s_i is an RNN hidden state for time i, computed by

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

It should be noted that unlike the existing encoder-decoder approach (see Eq. 1), here the probability is conditioned on a distinct context vector c_i for each target y_i.

The context vector c_i depends on a sequence $(h_1, \cdots, h_{T_x})$ to which an encoder maps the input sentence. Each token h_i contains information about the whole input sequence with a strong focus on the parts surrounding the i^{th} token of the input sequence. The context vector c_i is then computed as:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{3}$$

The weight α_{ij} of each h_j is computed by

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{4}$$

where

$$e_{ij} = a(s_{i-1}, h_j)$$

is an alignment model which scores how well the inputs around position j and the output at position i match. The score is based on the RNN hidden state s_{i-1} and h_j. The alignment model directly computes a soft alignment, which allows the gradient of the cost function to be backpropagated through. This gradient can be used to train the alignment model as well as the whole translation model jointly.

Let α_{ij} be the probability that the target token y_i is aligned to a source token x_j. Then, the i^{th} context vector c_i is the expected annotation over all the annotations with probabilities α_{ij}. The probability α_{ij}, or its associated energy e_{ij}, reflects the importance of h_j with respect to the previous hidden state s_{i-1} in deciding the next state s_i and generating y_i. Intuitively, this implements a mechanism of attention in the decoder. The decoder decides parts of the source sequence to pay attention to. By letting the decoder have an attention mechanism, we relieve the encoder from the burden of having to encode all information in the source sequence into a fixed length vector. With this approach the information can be spread throughout the sequence, which can be selectively retrieved by the decoder accordingly.

B. Bidirectional LSTM Encoder for pose correction

The usual RNN, described in Section 2, reads an input sequence x in order starting from the first symbol x_1 to the last one x_{T_x}. However, in the proposed scheme, we would like each word to summarize not only the preceding words, but also the following words. Hence, we use a bidirectional RNN, which has been successfully used recently in speech recognition (see, e.g., Graves *et al.* [20]).

For each token x_j, h_j is obtained by concatenating the forward hidden state $\vec{h}_j$ and the backward one $\overleftarrow{h}_j$. In this way, the h_j contains the summaries of both the preceding tokens and the following tokens. Due to the tendency of RNNs to better represent recent inputs, h_j will be focused on the words around x_j. This sequence is used by the decoder and the alignment model later to compute the context vector (Eqs. 3, 4).

We have initial pose estimates for each frame from the initialization stage. The correction model is trained to refine these sequences. We train the model to map the input pose sequence to target sequence (ground truth pose). There is a soft alignment between the input and output sequence elements. We now demonstrate the results of our approach on standard baselines and benchmark datasets.

IV. Experiments

We test our approach on two methods, SFN [2] and YR [1] on benchmark datasets. Experimental details are discussed below.

A. Datasets

YouTube Pose. This new dataset consists of 50 videos of different people from YouTube by [3], each with a single person in the video. Videos range from approximately $2,000$ to $20,000$ frames in length. For each video, 100 frames were randomly selected and manually annotated ($5,000$ frames in total). The dataset covers a broad range of activities, e.g., dancing, stand-up comedy, how-to, sports, disk jockeys, performing arts and dancing sign language signers.

YouTube Pose Subset. A five video subset from YouTube Pose. The videos distribution for subset dataset is as follows: two disc jockeys, a mime artist, a dancing sign language signer, and one aerobics instructor.

CVIT-Sports. For our experiments, we use the CVIT-SPORTS- videos dataset by [24]. It is an extremely challenging dataset of humans playing sports. This set has a total of 11 videos of a human playing sports retrieved from YouTube. It includes intricate activities like cricket-bowling, cricket-batting, football. In total, this set has a total of 1457 frames averaging out to 131 frames per video. All the frames in the dataset have been annotated with 14 key-points i.e full human pose. In our experiments, we use only the upper body joints.

These datasets vary in terms of activities, sampling rate, shape variance, and illumination. We demonstrate our experiments on a wide variety of datasets, which indicates the robustness of the proposed model.

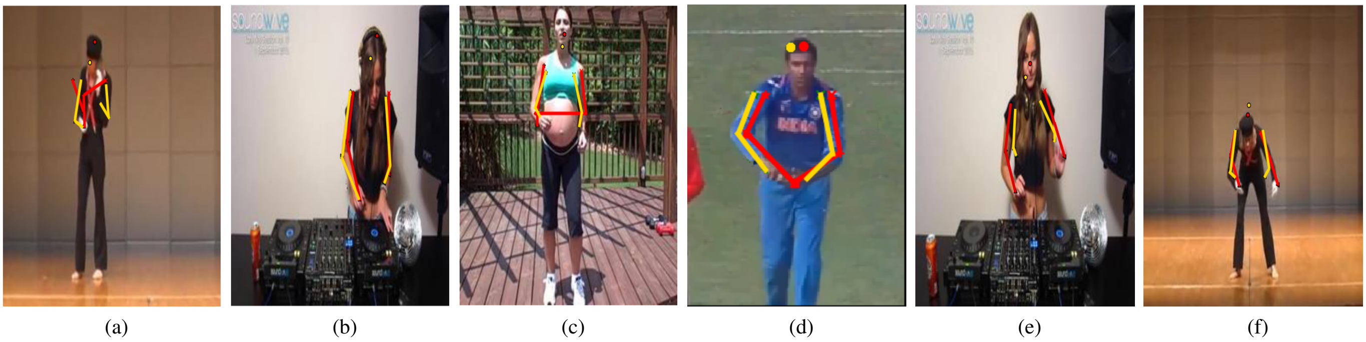

(a) (b) (c) (d) (e) (f)

Figure 3: **Comparing human poses on sample images from YouTube Pose Subset and CVIT Sports videos dataset.** (a), (b), (c), (d) show examples of pose corrections and (e), (f) show failure cases where red and yellow correspond to joints predicted using initialization (baseline) and our method respectively.

B. Baselines

SFN. SFN [2] is a state-of-the-art ConvNet for human pose estimation. It consists of a spatial ConvNet (8 convolution layers) and spatial fusion layers (5 convolution layers). It is a fully convolutional network with an implicit spatial model predicts a confidence heatmap for each body joint in images.

YR. [1] is a method for detecting articulated people and estimating their pose from static images based on a new representation of deformable part models. The flexible mixture model jointly captures spatial relations between part locations and co-occurrence relations between part mixtures, augmenting standard pictorial structure models that encode just spatial relations.

C. Evaluation Measures

In all the experiments, we compare the estimated joints against frames with manual ground truth. We present results as graphs that plot accuracy vs normalized distance from ground truth, where a joint is deemed correctly located if it is within a set threshold distance from a marked joint centre in ground truth. Higher pck implies more accurate estimations.

Training. The videos are split into fixed length sequences. To increase the total number of samples to train the model, we perform data augmentation. The frames are randomly rotated between $-30°$ and $30°$ and only horizontally flipped. Data augmentation has to be done carefully so that the video generated after augmentation should be semantically meaningful. By considering overlapping sequences, there are two-fold advantages: (i) this increases the number of samples for training, and (ii) overlapping sequences generate multiple estimates for a single frame, which reduces the total error.

The data is split into mini-batches of size 64. The correction model is trained on the YouTube Pose dataset. We used Keras for our experiments. For experiments on CVIT-SPORTS, the correction model is fine-tuned on a subset of CVIT-SPORTS videos dataset. The model is trained for 100 epochs, using RMSProp optimizer. The learning rate is set to 0.01.

Table I: **Component analysis on YouTube Subset Pose datasets.** Accuracy (%) at d = 20 pixels. SFN^{++} and YR^{++} indicates refinement using the proposed method. (We have highlighted all results where the proposed method shows improvement.)

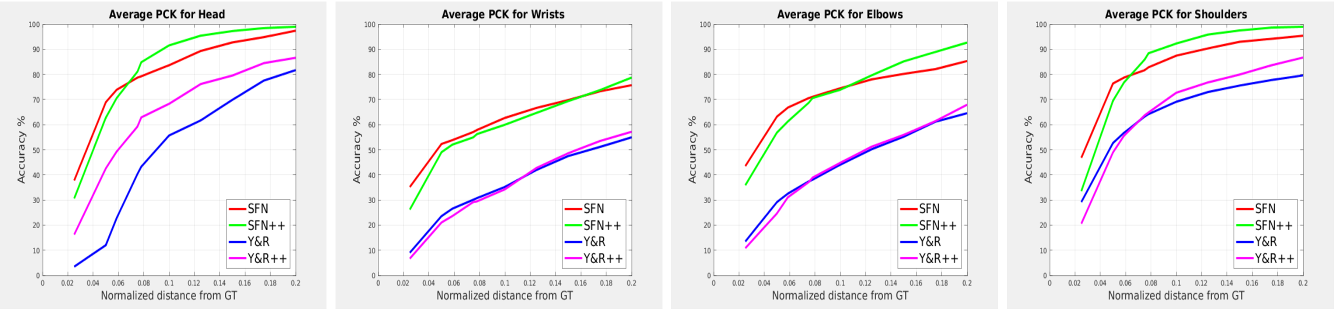
Method	Head	Wrsts	Elbws	Shldrs	Average
Pfister *et al.* [2]	74.4	59.0	70.7	82.7	71.3
SFN [2]	79.2	58.4	71.1	82.4	71.9
$\text{SFN}^{++}(Ours)$	**84.9**	56.8	71.0	**88.3**	**73.9**
YR [1]	44.6	30.3	37.9	64.1	44.0
$\text{YR}^{++}(Ours)$	**62.8**	29.4	**38.9**	**67.3**	**47.7**

Table II: **Component analysis on CVIT SPORTS videos dataset.** Accuracy (%) at d = 20 pixels. SFN^{++} and YR^{++} indicates refinement using the proposed method.(We have highlighted all results where the proposed method shows improvement.)

Method	Head	Wrsts	Elbws	Shldrs	Average
SFN [2]	20.7	46.7	38.9	55.7	43.2
$\text{SFN}^{++}(Ours)$	**46.9**	42.7	**38.9**	**56.7**	**46.3**
YR [1]	78.9	43.9	49.8	73	59.1
$\text{YR}^{++}(Ours)$	78.6	**44.0**	**51.2**	**74.3**	**59.8**

D. Results

The YouTube Pose Subset accuracy (%) at $d = 20$ pixels is shown in Table I. Our method surpasses SFN [2] by 2.07% . There is 5% boost in accuracy for head and shoulders improve by 6% (Table I). Our method performs equal to the baseline on wrists and elbows. Table I shows that we surpass YR [1] by 3.7%. We see that our method corrects head, elbows and shoulders but doesn't improve on wrists (which are harder to define in complex poses). We show 18%, 1%, 3% boost over YR on head, elbows and shoulders respectively. Charles *et al.* [3] mentioned that YR model doesn't perform well



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

Figure 4: **Results of our approach on YouTube Pose Subset dataset.** We observe that the refined estimates using our approach have higher recall compared to the baselines: Yang & Ramanan [1] and Spatial Fusion Network [2].

on the YouTube Pose dataset. Hence, they have re-trained the model to improve the estimates. Hence, we see that the YR average pck is 44.0% (ref table I) which is less compared to the accuracies mentioned in [3]. Experiments demonstrate that the proposed approach refines predictions given generic pose estimates.

The PCK (Percentage of Correct Keypoints) accuracy on the CVIT-SPORTS videos dataset is shown in Table II. We see improvement in head, elbows and shoulders over SFN. The head joint accuracy enhances by 26.2%. The accuracy averaged over all joints exceeds baseline by 3.1%. While using YR baseline, there is boost in wrists, elbows and shoulders and the average pck gain is 0.7%.

Figures 3 and 1 show the visualizations of pose corrections on sample dataset images. The red and yellow represent the initialization (baseline predictions) and the corrected pose predictions respectively. In Figure 3(a) and (b), the initial predictions for left elbow and left wrist are erroneous, but our approach corrects the poses as shown in the figure. Figure 3(c) predicts left wrist on the right wrist while Figure 3(d) predicts right wrist on the left wrist, and our method successfully corrects the pose. Our method fails to correct the poses, if the initial predictions are erroneous across the neighborhood. In Figures 3(e) and (f), it is not able to refine the pose well enough, as the neighborhood frames also have erroneous predictions, which makes it difficult for refinement. Adding to that, these videos have low sampling rate and large motion changes across frames. For example, Figure 3(f) is the only frame where the head position lies in center but its previous and next frames have head position in the top (as shown in Figure 3(a)) and this leads to the errors.

The PCK plots for head, wrists, elbows and shoulders on the YouTube Pose Subset dataset are shown in Figure 4. It is clear from the figure that the proposed approach has high recall. Also, the gain in accuracy is highest for head, followed by shoulders, elbows and wrists. Higher recall is an indication of refinement in joint predictions (Figure 1and 3).

V. Conclusion

In this paper, we showed that the proposed pose correction model refines pose estimates obtained from generic models, independent of the pose estimator used to generate the initial pose estimates. We successfully posed the pose correction problem as a sequence-to-sequence learning problem. We demonstrated our results on challenging datasets which cover a wide range of activities, and are sampled at different sampling rates. The results show great promise in this approach to get more accurate pose estimation results in a simple, fast and generalizable manner.

References

[1] Y. Yang and D. Ramanan, “Articulated pose estimation with flexible mixtures-of-parts,” in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR ’11. Washington, DC, USA: IEEE Computer Society, 2011.

[2] T. Pfister, J. Charles, and A. Zisserman, “Flowing convnets for human pose estimation in videos,” in *IEEE International Conference on Computer Vision*, 2015.

[3] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman, “Personalizing human video pose estimation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[4] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition,” *Int. J. Comput. Vision*, 2005.

[5] M. Andriluka, S. Roth, and B. Schiele, “Pictorial structures revisited: People detection and articulated pose estimation.” in *CVPR*, 2009.

[6] B. Sapp, A. Toshev, and B. Taskar, *Cascaded Models for Articulated Pose Estimation*, 2010.

[7] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari, “2d articulated human pose estimation and retrieval in (almost) unconstrained still images,” *International Journal of Computer Vision*, 2012.

[8] M. Eichner and V. Ferrari, “Better appearance models for pictorial structures,” in *Proceedings of the British Machine Vision Conference*, 2009.

[9] D. Ramanan, “Learning to parse images of articulated bodies,” in *Advances in Neural Information Processing Systems 19*, 2007.

[10] X. Chen and A. Yuille, “Articulated pose estimation by a graphical model with image dependent pairwise relations,” in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[11] A. Jain, J. Tompson, Y. LeCun, and C. Bregler, “Modeep: A deep learning framework using motion features for human pose estimation,” *CoRR*, 2014.

[12] N. Jammalamadaka, A. Zisserman, and C. V. Jawahar, “Human pose search using deep poselets,” in *International Conference on Automatic Face and Gesture Recognition*, 2015.

[13] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman, “Deep convolutional neural networks for efficient pose estimation in gesture videos,” in *Asian Conference on Computer Vision (ACCV)*, 2014.

[14] A. Toshev and C. Szegedy, “Deeppose: Human pose estimation via deep neural networks,” in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[15] J. J. Tompson, A. Jain, Y. Lecun, and C. Bregler, “Joint training of a convolutional network and a graphical model for human pose estimation,” in *Advances in Neural Information Processing Systems 27*, 2014.

[16] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, “Sequential deep learning for human action recognition,” in *Proceedings of the Second International Conference on Human Behavior Unterstanding*, 2011.

[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR ’14, 2014.

[18] N. Kalchbrenner and P. Blunsom, “Recurrent continuous translation models,” 2013.

[19] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Oct. 2014.

[20] A. Graves, “Generating sequences with recurrent neural networks,” *CoRR*, 2013.

[21] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *CoRR*, 2014.

[22] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML ’06, 2006.

[23] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, ser. NIPS’14, 2014.

[24] D. Singh, V. Balasubramanian, and C. Jawahar, “Finetuning human pose estimations in videos,” in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, 2016.