P3DAttnNet: Automated Assembly Plan Generation from Video Demonstration

Abhinav Upadhyay
Accenture Labs, Bangalore
k.a.abhinav@accenture.com

Priyanshu Abhijit Barua
Indian Institute of Information Technology, Pune
baruapriyanshu@gmail.com

Alpana Dubey
Accenture Labs, Bangalore
alpana.a.dubey@accenture.com

Shubhashis Sengupta
Accenture Labs, Bangalore
shubhashis.sengupta@accenture.com

Suma Mani Kuriakose
Accenture Labs, Bangalore
suma.mani.kuriakose@accenture.com

Piyush Goenka
Accenture Labs, Bangalore
piyush.goenka@accenture.com

***Abstract*—In this work, we propose a deep neural network, P3DAttnNet, for automatically generating assembly plans from video demonstrations. We develop a spatio-temporal attention model to recognize actions from the video. We apply a functional object-oriented network to model the assembly plan. We evaluate our network on IKEA ASM dataset consisting of 371 unique assemblies having 1113 RGB videos and 371 depth videos. We perform quantitative analysis along three metrics - Frame accuracy, Macro-recall, and Mean average precision (mAP) to evaluate the efficacy of our approach. We compare our approach with the existing baseline and significantly outperform on all these metrics.**

I. Introduction

Cellular manufacturing requires considerable labor and is significantly dependent on manpower. To enable robots to perform such manufacturing tasks without providing detailed instructions of the task, a robot must be able to automatically understand the details of the task and accomplish it based on simple or ambiguous knowledge of the task. Video demonstrations are usually available for the assembly of certain products, such as furniture. There is a need to understand the assembly instructions from such videos.

Video analysis has many real-world applications, including behavior analysis, video retrieval, human-robot interaction, gaming, and entertainment [14]. Over the last decade, there has been a growing interest in video action recognition with the emergence of high-quality, large-scale action recognition datasets [8][14] and advancements in deep learning based approaches [2][5][6]. Generating assembly or disassembly sequences from instructional videos is still an active area of research [11].

In this work, we propose a deep neural network, P3DAttnNet, for automatically generating assembly plans from video demonstrations. We develop a spatio-temporal attention model to recognize actions performed in a video. We apply a functional object-oriented network to model the assembly plan graph. We evaluate our network on the IKEA ASM dataset consisting of 371 unique assemblies having 1113 RGB videos and 371 depth videos. We perform quantitative analysis along three metrics - Frame accuracy, Macro-recall, and Mean average precision (mAP) to evaluate the efficacy of our approach. We compare our approach with the existing baseline and significantly outperform on all these metrics.

II. Related work

Synthesizing an assembly plan from a video demonstration is an active area of research. Many studies have been conducted to construct assembly sequences based on the geometrical relationships of assembly parts. Paulius et al. [4][9][10] model cooking instructions in the form of a graph based on a video demonstration. They propose a graph based knowledge representation called Functional Object-Oriented Network (FOON) to denote a set of task instructions. The approach provides a structured knowledge representation which is constructed from observations of human activities and manipulations. FOON expresses the relationship between motions and objects, and the change in the object state owing to those motions. They manually construct these functional units by labelling instructional videos. Our work is inspired from [11], where they generates an Assembly Task Sequence Graph (ATSG) by recognizing a graphical instruction manual. An ATSG is a graph describing the assembly task procedure by detecting types of parts included in the instruction images. Hussein et al. [1] propose a graph based network to recognize minutes-long human activities consisting of a set of unit-actions. Aksoy et al. [12] generate a textual description for an unseen complex from the video. Nicolescu et al. [13] propose a framework for directly mapping a complex verbal instruction to an executable task representation, from a single training experience. Jonathan et al. [15] generate a graphical representation by recognizing activities from assembly videos in the form of change in kinematic state of objects. The change in states predicted by the model is generalised in the form of connect or disconnect. They test the model on egocentric videos of an Ikea chair and third person block assembly RGBD videos with inertial measurements. Carreira et al. [2] propose a Two-Stream Inflated 3D ConvNet (I3D) that is based on 2D ConvNet inflation to learn seamless spatio-temporal feature extractors from videos. Unlike prior approaches, our work focuses on recognizing a complex assembly sequence from a video demonstration. We extract information about

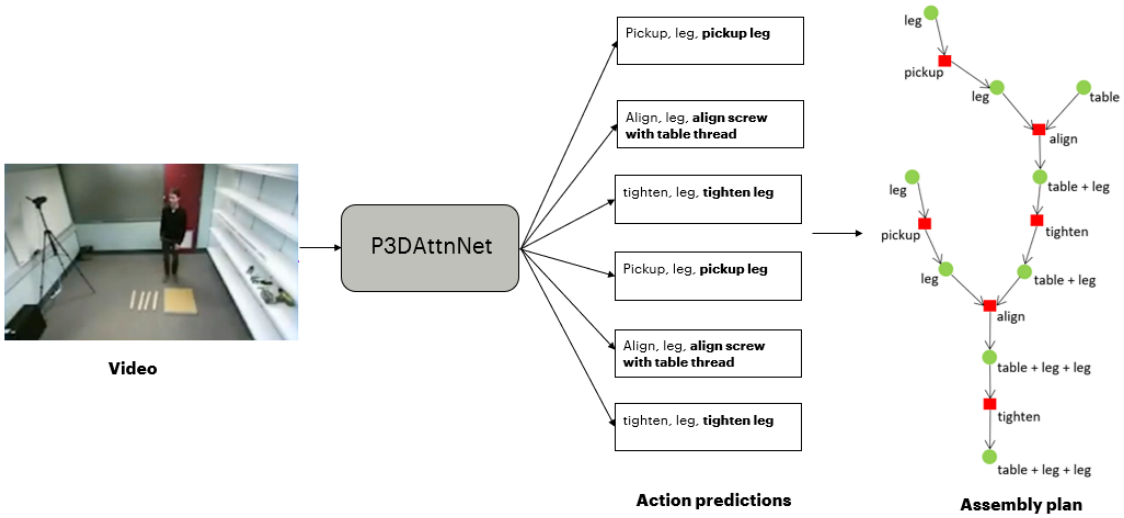

Fig. 1: Actions predicted by our network from a video demonstration (clipped video). The predicted actions are used to generate the assembly plan.

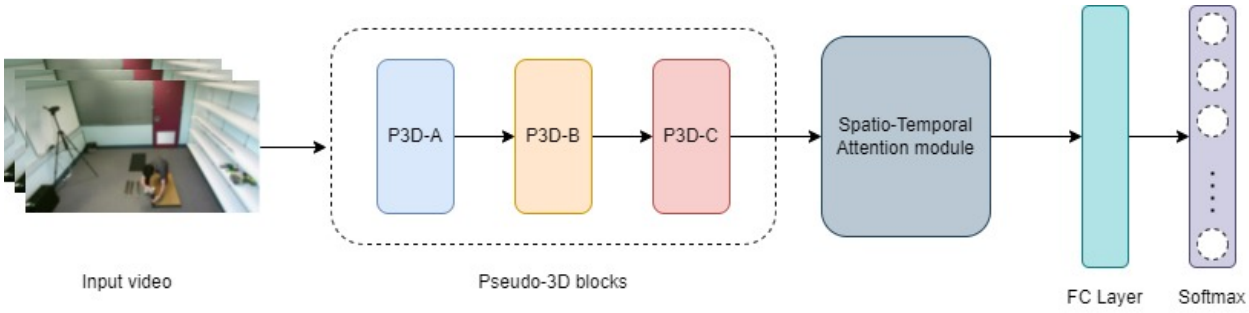

Fig. 2: The input video is fed to the Pseudo 3D blocks to generate the feature maps. Then, we apply spatio-temporal attention model to automatically learn the discriminative regions from each image frame. The network predicts per frame action labels.

the assembly parts from the instructional video and estimate several feasible actions needed for the assembly task.

III. Approach

Our goal is to generate an assembly plan from a video demonstration. We apply 3D convolutions [6] to encode the spatio-temporal information. 3D convolutions simultaneously model the spatial information like 2D filters and build temporal connections across frames within the video. 3D convolutional filters are represented as d * k * k where d is the temporal depth of the kernel and k is the kernel spatial size. We use a Pseudo 3D (P3D) block to extract the feature representations from a video. In a Pseudo 3D block, the 3D convolutional filters (with size of 3*3*3) is decoupled into 1*3*3 convolutional filters equivalent to 2D CNN on spatial features and 3*1*1 convolutional filters equivalent to 1D CNN for temporal features.

P3D has three residual blocks (P3D-A, P3D-B, and P3D-C) designed based on (a) whether the modules of 2D filters on spatial dimension (S) and 1D filters on temporal domain (T) should directly or indirectly influence each other and (b) whether the two kinds of filters should both directly influence the final output. The three residual blocks are shown in Figure 3.

P3D-A: In this block, the temporal block follows the spatial block in a cascading manner. The two blocks directly influence each other and only the temporal block is connected to the final output.

P3D-B: In this block, the spatial and temporal blocks are connected in parallel and indirectly influence each other. Both the blocks are connected to the final output.

P3D-C: In this block, the temporal block follows the spatial block in a cascading manner. Both the blocks are connected to the final output.

We apply a spatio-temporal attention model to automatically learn the discriminative regions from each image frame. The network architecture is shown in Figure 4. The spatio-temporal

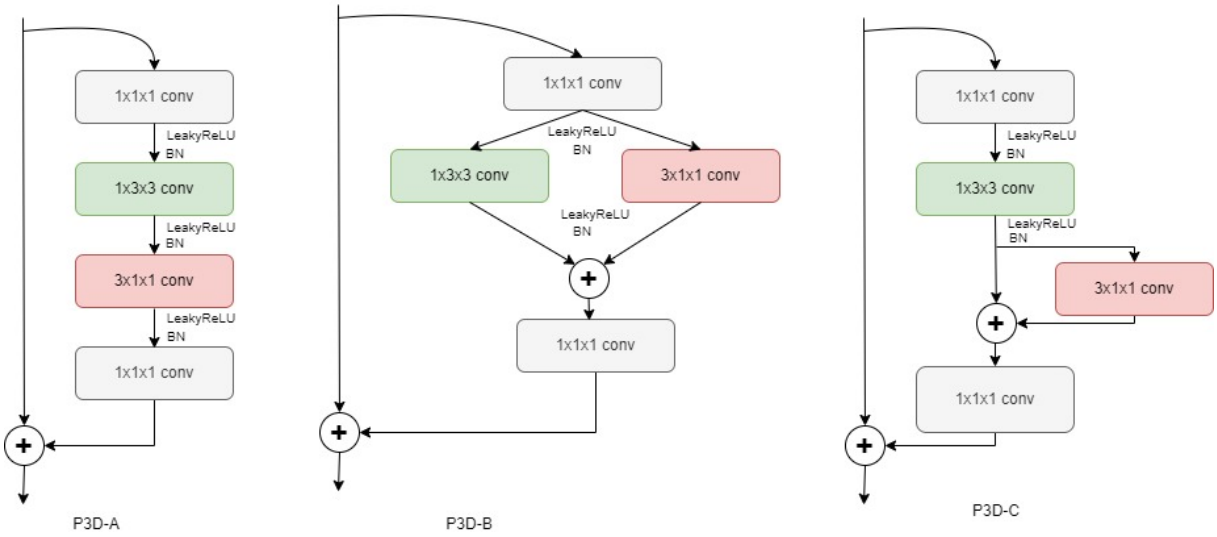

Fig. 3: Pseudo 3D Blocks

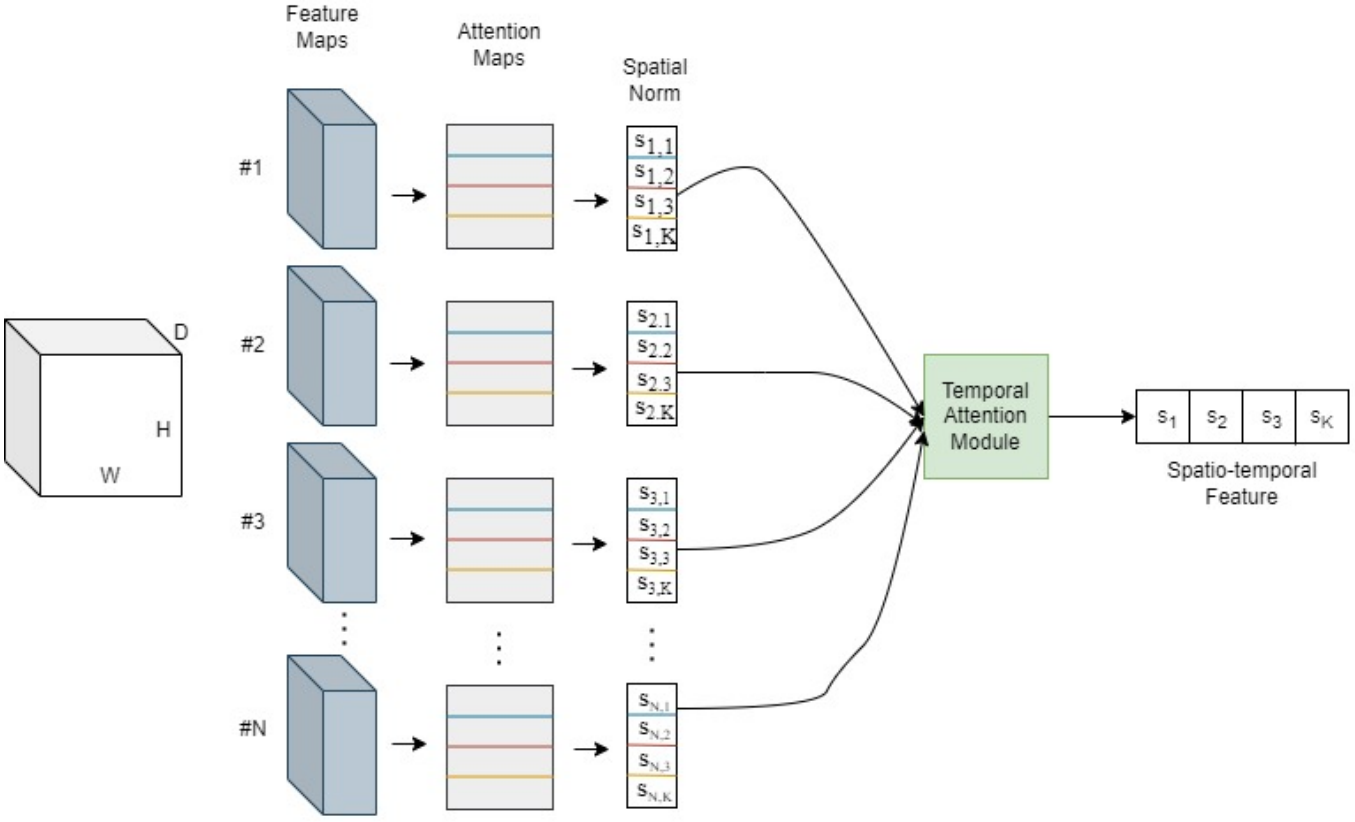

Fig. 4: **Spatio-temporal Attention module**: Given a set of feature maps from the input video, we generate corresponding attention maps for each frame. The attention maps are horizontally split into K blocks. We apply L2 norm to compute attention score for each spatial region of different frames. Temporal attentions compute an aggregate representation for the set of features generated by each spatial attention model. Finally, the spatio-temporal features are concatenated into a single feature which represents the information contained in the entire video sequence.

attention block assigns attention weights to each spatial region in different frames. These attention weights include both spatial and temporal attention information. This helps in recognizing the discriminative regions and thereby frame

TABLE I: Action verb, object name and action description from IKEA ASM dataset [8]

Id	Verb	Object	Description
0	-	-	No Annotation (NA)
1	align	leg	align leg screw with table thread
2	align	side panel	align side panel holes
3	attach	back panel	attach drawer back panel
4	attach	side panel	attach drawer side panel
5	attach	shelf	attach shelf to table
6	flip	table	flip table
7	flip	shelf	flip shelf
8	flip	table top	flip table top
9	insert	pin	insert drawer pin
10	lay down	back panel	lay down back panel
11	lay down	bottom panel	lay down bottom panel
12	lay down	front panel	lay down front panel
13	lay down	leg	lay down leg
14	lay down	shelf	lay down shelf
15	lay down	side panel	lay down side panel
16	lay down	table top	lay down table top
17	-	-	other (unavailable action class)
18	pick up	back panel	pick up back panel
19	pick up	bottom panel	pick up bottom panel
20	pick up	front panel	pick up front panel
21	pick up	leg	pick up leg
22	pick up	pin	pick up pin
23	pick up	shelf	pick up shelf
24	pick up	side panel	pick up side panel
25	pick up	table top	pick up table top
26	push	table	push table
27	push	table top	push table top
28	rotate	table	rotate table
29	slide	bottom panel	slide bottom of drawer
30	spin	leg	spin leg
31	tighten	leg	tighten leg
32	position	drawer	position the drawer right side up

selection. Given the feature maps of video $f = f_1, f_2, ..., f_N$, we compute corresponding attention map g_n by performing $L2$ normalization on the square sum through the depth channel.

$$g_n(h,w) = \frac{||\sum_{d=1}^{D} f_n(h,w,d)^2||_2}{\sum_{h,w}^{H,W} ||\sum_{d=1}^{D} f_n(h,w,d)^2||_2} \tag{1}$$

where H and W are the height and the width of the feature maps. Each frame has a corresponding attention map. The feature map is divided into K blocks horizontally each having corresponding attention maps.

$$g_n = [g_{n,1}, g_{n,k}..., g_{n,K}] \tag{2}$$

$$f_n = [f_{n,1}, f_{n,k}..., f_{n,K}] \tag{3}$$

where $g_{n,k}$ represents the spatial attention map of the k^{th} region of the n^{th} frame. Then, we apply $L1$ normalization on all values in each block to obtain one spatial attention score for that region.

$$s_{n,k} = \sum_{i,j} ||g_{n,k}(i,j)||_1 \tag{4}$$

The same procedure is followed on all the selected frames of the input video to obtain the N * K matrix S of spatial attention scores.

All the parts of an object are not clearly visible in every video frame because of self-occlusion or an explicit foreground occluder. Therefore, pooling features across time using a per-frame weight α_n is not sufficiently robust, since some frames could contain valuable partial information (e.g. other parts for context). Instead of applying the same temporal attention weight α_n to all features extracted from a frame n, we apply multiple temporal attention weights $\alpha_{n,1}, \ldots, \alpha_{n,K}$ to each frame, i.e one for each spatial component. With this approach, the temporal attention model is able to assess the importance of a frame based on the relevance of the different salient regions.

We define the temporal attention $\alpha_{n,K}$ for the spatial component k in frame n to be the softmax of a linear function

$$e_{n,k} = (w_{t,k})^T . s_{n,k} + b_{t,k} \tag{5}$$

where $s_{n,k} \in R^D$ is the feature of the k^{th} spatial component in the n^{th} frame, and $w_{t,k} \in R^D$ and $b_{t,k}$ are parameters to be learned.

The temporal attention model directly computes a soft attention weight for each frame. The importance weight $\alpha_{n,k}$ for each frame is

$$\alpha_{n,k} = \frac{e_{n,k}}{\sum_{j=1}^{N} e_{j,k}} \tag{6}$$

The weighting mechanism decides the importance of a frame based on the spatial regions. The temporal attentions are used to enhance spatial features by weighted averaging

$$s_k = \sum_{n=1}^{N} \alpha_{n,k} s_{n,k} \tag{7}$$

Finally, the entire input video is represented by a feature vector $s \in R^{K*D}$ generated by concatenating the temporal features of each spatial component

$$s = [s_1, ..., s_K] \tag{8}$$

Generating Assembly Instructions: To generate the assembly plan, we leverage the concept of functional object-oriented network (FOON) [4]. A FOON represents a graph for several activities. It is a sequence of functional units that captures information on the objects, manipulations and actions required to fulfill the task's goal. It consists of two types of nodes in its bipartite structure: object nodes and motion nodes. Object nodes refer to objects that are used in activities, e.g. leg, side panel, shelf, and drawer, while motion nodes refer to actions that can be performed on said objects such as flip shelf or push table. A FOON is a directed graph, as some nodes are the outcomes of the interaction between other nodes. An edge, denoted as E, connects two nodes. Edges are drawn from either an object node to a motion node, or vice-versa.

The proposed assembly plan is a graphical representation of all object interactions and the associated actions observed in the assembly video. It is a directed graph where each node represents an object (a part of the product being assembled) or an action (a motion acting on one or two objects causing a

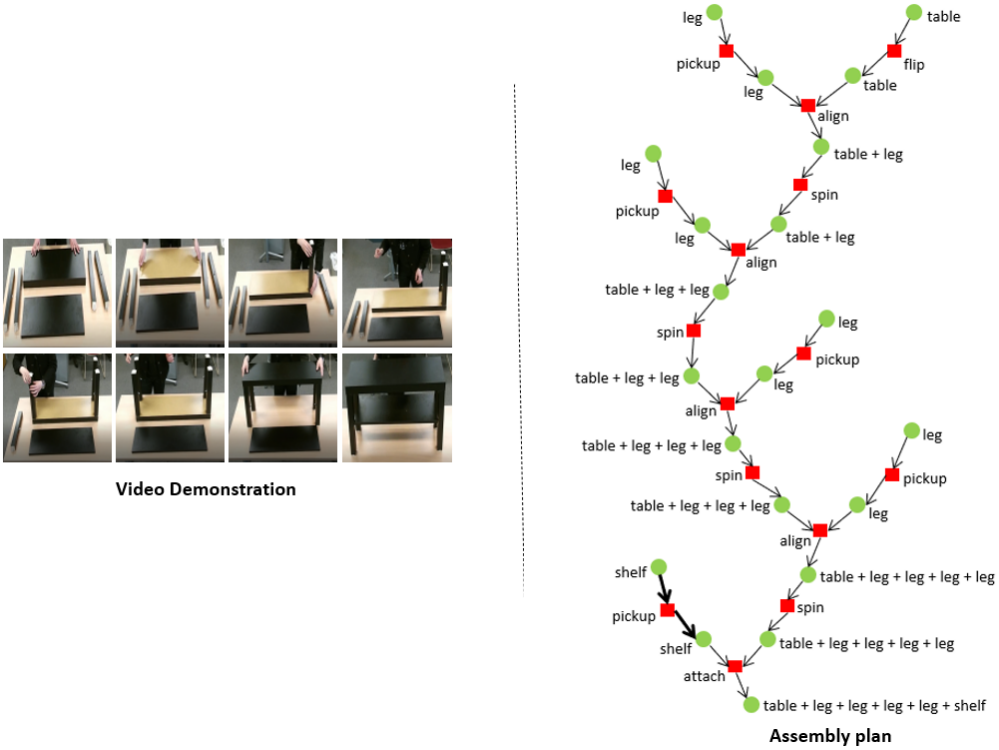

Fig. 5: The generated assembly plan from our approach for TV Bench (partly shown here). The highlighted arrows indicate instructions that our model was not able to predict from the video. These instructions have been added manually for the sake of completion.

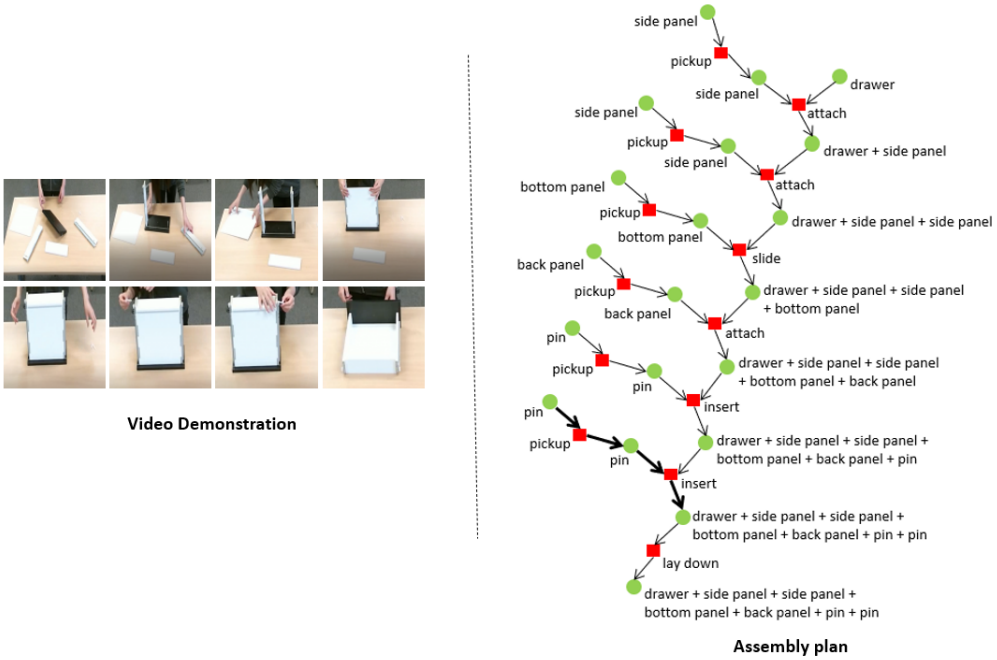

Fig. 6: The generated assembly plan from our approach for TV Bench (partly shown here). The highlighted arrows indicate instructions that our model was not able to predict from the video. These instructions have been added manually for the sake of completion.

change in the state of the object). We use a list similar to an adjacency list for a graph, to represent our assembly plan. Each element of the list represents a single action acting on an object resulting in a new state for the object. Generating the assembly plan involves - (a) filtering the output of the action recognition model by removing low confidence predictions (we choose a threshold of 0.5) and (b) tracking each object based on a list of objects and the effects of different actions on these objects.

TABLE II: Performance of our approach along three metrics and comparison with baseline

Approaches	Frame accuracy		Macro-recall	mAP
	Top 1	Top 3		
ResNet18 [7][8]	27.06	55.14	21.95	11.69
ResNet50 [7][8]	30.38	56.1	20.03	9.47
C3D [5][8]	45.73	69.56	32.48	21.98
I3D [2][8]	57.57	76.55	39.34	28.59
P3D [6][8]	60.4	81.07	45.21	29.86
P3DAttnNet (ours)	**69.23**	**87.48**	**56.75**	**43.11**

IV. Evaluation and Results

A. Dataset

The IKEA ASM dataset consists of 371 unique assemblies of four different furniture types (side table, coffee table, TV bench, and drawer) in three different colors (white, oak, and black) [8]. There are 1113 RGB videos and 371 depth videos (top view) in total. The dataset contains 3,046,977 frames (∼35.27h) of footage with an average of 2735.2 frames per video (∼1.89min). The dataset contains a total of 16,764 annotated actions with an average of 150 frames per action (∼6sec). Table I shows the list of objects and atomic actions in the IKEA ASM dataset.

B. Experimental-setup

We use Adam optimizer with an initial learning rate of 0.001. We train our model for 300 epochs with a batch size of 10. We use LeakyReLU activation. We split our dataset into train and test set consisting of 254 and 117 assembly demonstrations respectively.

C. Results

We evaluate our approach for action recognition using three metrics [8]: (i) **Frame-wise accuracy (FA)**: Fraction of the number of correctly classified frames to the total number of frames in each video, averaged over all videos in the test set, (ii) **Macro-recall**: As the dataset is imbalanced, we report the macro-recall by separately computing recall for each category and then averaging it, and (iii) **Mean average precision (mAP)**: As all the videos contain multiple action labels, we compute mean Average Precision. The mean Average Precision (mAP) score is calculated by taking the mean AP over all classes. Table II shows the performance of our approach with respect to these metrics on the IKEA ASM [8] dataset.

We observe that our model fails to predict instructions accurately from video frames under the following circumstances:

1) The human makes a mistake or is confused while performing the action.
2) The object involved in the action is very small or occluded (as in the case of a drawer pin).
3) The human performs the action too quickly and jumps to another action.

Comparison: We compare our approach with state-of-the-art methods - C3D [5], I3D [2], P3D [6], and Frame-wise ResNet [7], for action recognition along these three metrics. Our approach significantly outperforms the existing work across all the metrics (as shown in Table II). We observe a relative increase of 14.6% in Frame accuracy (Top 1), 25.5% in Macro-recall, and 44.37% in mAP (on test set) in comparison with P3D (existing baseline).

The assembly plans for TV Bench and Shelf Drawer generated by our approach (from test dataset) are shown in Figure 5 and Figure 6 respectively.

V. Conclusion

We propose a deep neural network, P3DAttnNet, for automatically generating assembly plans from video demonstrations. We develop spatio-temporal attention model to recognize actions from a video. We evaluate our approach on the IKEA ASM dataset consisting of 371 unique assemblies having 1113 RGB videos and 371 depth videos. We compare our approach with the existing baseline and significantly outperform on three metrics. As future work, we plan to integrate our system with robotic assembly planning, where robots can assemble IKEA furniture based on the instructions generated by our approach.

References

[1] Hussein, N., Gavves, E., & Smeulders, A. W. (2019). Videograph: Recognizing minutes-long human activities in videos. arXiv preprint arXiv:1905.05143.

[2] Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6299-6308).

[3] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19).

[4] Paulius, D., Huang, Y., Milton, R., Buchanan, W. D., Sam, J., & Sun, Y. (2016, October). Functional object-oriented network for manipulation learning. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 2655-2662). IEEE.

[5] Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 4489-4497).

[6] Qiu, Z., Yao, T., & Mei, T. (2017). Learning spatio-temporal representation with pseudo-3d residual networks. In proceedings of the IEEE International Conference on Computer Vision (pp. 5533-5541).

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[8] Ben-Shabat, Y., Yu, X., Saleh, F., Campbell, D., Rodriguez-Opazo, C., Li, H., & Gould, S. (2021). The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 847-859).

[9] Paulius, D., Jelodar, A. B., & Sun, Y. (2018, May). Functional object-oriented network: Construction & expansion. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5935-5941). IEEE.

[10] Paulius, D., Dong, K. S. P., & Sun, Y. (2019). Functional object-oriented network: Considering robot's capability in human-robot collaboration. arXiv preprint arXiv:1905.00502, 16.

[11] Sera, I., Yamanobe, N., Ramirez-Alpizar, I. G., Wang, Z., Wan, W., & Harada, K. (2021, June). Assembly Planning by Recognizing a Graphical Instruction Manual. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 3138-3145). IEEE.

[12] Aksoy, E. E., Ovchinnikova, E., Orhan, A., Yang, Y., & Asfour, T. (2017). Unsupervised linking of visual features to textual descriptions in long manipulation activities. IEEE Robotics and Automation Letters, 2(3), 1397-1404.

[13] Nicolescu, M., Arnold, N., Blankenburg, J., Feil-Seifer, D., Banisetty, S., Nicolescu, M., ... & Monteverde, T. (2019, October). Learning of complex-structured tasks from verbal instruction. In 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids) (pp. 747-754). IEEE.

[14] Zhu, Y., Li, X., Liu, C., Zolfaghari, M., Xiong, Y., Wu, C., ... & Li, M. (2020). A comprehensive study of deep video action recognition. arXiv preprint arXiv:2012.06567.

[15] Jones, J. D., Cortesa, C., Shelton, A., Landau, B., Khudanpur, S., & Hager, G. D. (2021). Fine-grained activity recognition for assembly videos. IEEE Robotics and Automation Letters, 6(2), 3728-3735.