JoTyMo: Joint type and movement detection from RGB images of before and after interacting with articulated objects

1st Ehsan Forootan
ECE department (University of Tehran)
Human and Robot Interaction Laboratory
Tehran, Iran
esa.forootan@ut.ac.ir

2nd Hamed Ghasemi
ECE department (University of Tehran)
Human and Robot Interaction Laboratory
Tehran, Iran
hamed.ghasemi@ut.ac.ir

3rd Mehdi Tale Masouleh
ECE department (University of Tehran)
Human and Robot Interaction Laboratory
Tehran, Iran
m.t.masouleh@ut.ac.ir

***Abstract*—This paper presents a CNN-based network architecture aimed at the classification and detection of joint types within articulated objects, specifically the Push-P joints, P-joints, and R-joints. Additionally, it aims to distinguish these joints from objects lacking interactable components. The movement modeling of the detected articulated objects is explored by analyzing pre and post interaction RGB images within the SAPIEN PartNet-Mobility Data set. To achieve this, an architecture is proposed to leverage consecutive CNN encoders based on the VGG architecture, in order to classifying joints based on pre and post-interaction images. Additionally, in order to predict the effect point and movement vector a convolutional encoder is considered for each joint type. Moreover, extensive evaluation showcases notable results, achieving 96% accuracy in joint classification and 94% accuracy in regression on the considered dataset. Furthermore, this study presents an in-depth evaluation of architecture's performance in comparison to traditional techniques such as SVM and random forests, as well as transfer learning methods utilizing ResNets and VGG-16 networks. The results underscore the accuracy achieved by JoTyMo in both classification and regression tasks, along with its ability to generalize effectively across unseen simulated objects in PartNet-Mobility dataset, in contrast with other methods.**

I. INTRODUCTION

Grasping can be regarded as one of the fundamental but most challenging skills of robots. Grasping is also important for enabling robotics arms to perform various tasks in different environments. One of the main challenges of robotic grasping is how to deal with objects which have restrictions in movement such as articulated objects. These articulated objects comprise three distinct types of joints: P-joints, PushP-joints, and R-joints. P-joints possess a prismatic element that can be closed or opened. In contrast, PushP-joints are limited to either retracting or opening, with one degree of freedom restricted in comparison to P-joints. Meanwhile, R-joint objects incorporate rotational joints, enabling rotation around a central point. Due to these structural differences and the necessity to differentiate them from non-articulated objects, a classification based on observed interaction is warranted. Furthermore, to model the movement of any of these joints, two points in space are required: one indicating the point where force is being applied and the other indicating the resulting position of the object. Consequently, this necessitates regression analysis, where points are regressed based on the observed interaction in pre-interaction and post-interaction images.

There are different approaches to tackle the problem of detecting and modeling articulated objects, such as model-free method suggested in [11], where Deep Reinforcement Learning (DRL) is used. Model-based methods, which use a model of the object and its joint parameters similar to [12], or data-driven learning-based methods, which use neural networks to learn the joint state from data have also been reported in the literature.

Using the Convolutional Neural Networks(CNNs) in articulated object modeling is a challenging problem due to the number of reasons, such as estimating the position and orientation of the joint axis and the current rotation angle of the object. However, works such as multi-camera video tracking and recognition [10], which utilize CNN to track objects in images and classify them, promote the usage of CNN in classifying images due to their success in achieving high precision in model recognition. Additionally, CNN-based learning methods, due to their reliance on data and experience rather than a fixed or predefined model of the environment, offer enhanced flexibility and improved generalization capabilities. This may allow them to adapt to changing or uncertain situations, and to handle complex or nonlinear dynamics which might be difficult to model [4].

They can also leverage existing frameworks, such as Keras [35], which have been proven to be effective in various tasks like image classification, object detection, and natural language processing. In fact, CNN-based learning methods can automatically learn relevant features from raw data and effectively capture the hierarchical patterns and structures within the input. Furthermore, CNN-based learning methods have the advantage of being able to be scaled to large datasets and handle high-dimensional data efficiently, as presented in [5]. Subsequently, Methods based on CNNs such as DEEP-SEE [1], CNN-based Joint State Estimation [2] and Single-

view 3D Open-able Part Detection [3] have been suggested for the task of detecting and modeling articulated objects in both real and simulated environments. DEEP-SEE utilizes two frames, one from the present moment and the other from the preceding instant, to specify the target for tracking and the corresponding search area. This approach has served as the catalyst for the primary contribution presented in this paper: the development of an architectural framework capable of detecting and modeling various types of joint movements.

This paper presents a CNN-based approach that utilizes two RGB images before and after an interaction on an object to identify joint types and movements of articulated objects, similar to the Open-able Part Detection (OPD) method described in [3]. However, unlike OPD, the approach uses two RGB images instead of one RGBD image, in order to ensure that the limitations of a joint are taken into consideration.The method presented in this paper focuses solely on the observed interaction and does not consider camera angle or mounting point, as long as the interaction with the joints is visible. This is in contrast to the suggested approach in [2], where the field of view of the cameras is limited by the robot and with [1], where type is assumed based on the object rather than the movement. Furthermore, proposed method is trained and tested on the PartNet-Mobility Dataset [6], which utilizes images captured both before and after interactions or view changes. While other datasets such as Shape2Motion [13] and ShapeNet-Core [14] exist, the method joint type and movement detection approach called JoTyMo, was exclusively trained on the Part-Net mobility dataset.The choice of selecting PartNet-Mobility Dataset was made because of the simulated environment objects it contains and the potential to generate numerous unseen data for each object. To visually represent this, Fig. 1 showcases several examples of these images. In addition, the objects in PartNet-Mobility Dataset are systematically classified into four distinct joint types: P-joint, R-joint, Push-P joint, and Not articulated. Further more, these joint types are defined based on Computer-Aided Design (CAD) but were used as RGB images for the purpose of this paper, to simplify and reduce networks sizes.

After the joint type is detected utilizing the first half of the network, illustrated in Fig. 2, an additional CNN architecture is employed in order to train on specific joint types. This facilitates the estimation of the movement vector and the joint effect point of the object post-classification. The movement vector signifies the manner in which the interaction occurred, resulting in the joint's motion. Similarly, the effect point indicates the location at which the force that generated the movement exerted its influence. Additionally, in order to assess JoTyMo's accuracy in both detecting the right joint type and detecting the movement, the obtained results from the JoTyMo model are compared with various classical machine learning algorithms such as random forests and transfer learning networks such as VGG-16.

The paper at hand is composed of 5 distinct parts. The first part consists of the introduction, which focuses on describing the problem for which an approach is being proposed. This is based on previous research and includes important terminology related to JoTyMo, such as joint types and their meanings, movement detection, effect point, data sets containing articulated objects, and the CNN-based approach. The next part provides a concise review of the related works upon which JoTyMo builds, as found in the literature. Following that, section III delves into the methodology and approach employed by JoTyMo. Part IV examines the achievements and comparisons of JoTyMo to other methods, showcasing its capabilities and limitations. Lastly, part V concludes by presenting a summary of findings, and suggests potential avenues for further improvement of JoTyMo.

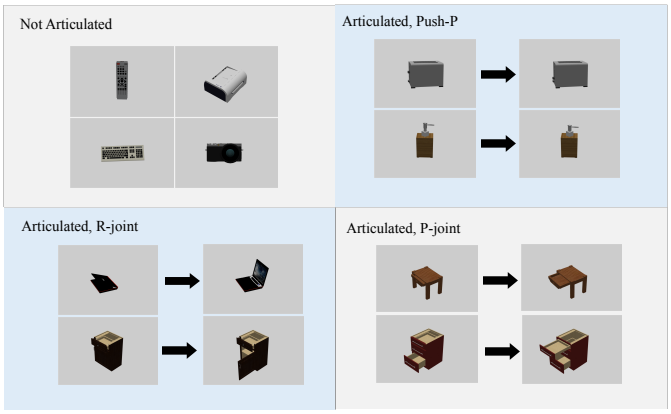

Fig. 1. Different examples of distinct objects along with their potential interactions within the PartNet-Mobility Dataset [6].

II. Related Works

This section will discuss the works that are relevant to JoTyMo. Part A is about works related to JoTyMo which delve into the classification of images based on images. Moreover, works related to regression in images are briefly mentioned. After that, in part B, literature that utilizes CNNs to estimate joint pose and movement will be shown, followed by a summary of an effective previous approach which inspired JoTyMo. Part C demonstrate where the inspiration for JoTyMo's architecture for modeling articulated objects was derived from. Lastly, in part D, some of the works which JoTyMo intends to follow in order to model articulated objects will be illustrated.

A. Deep Convolutional Networks for Image Classification and Regression

Various architectures have been proposed for image classification, including VGG [8], Alexnet [9], Inception-V3 [23], and ResNets [22], [23]. For the purpose of this study, the VGG-based design was selected due to its use of smaller kernel sizes. To the end of addressing image-to-image regression, a method was developed in [25], where architecture is fully convolutional and can handle multiple images during inference and obtains comparable results without any post processing or task-specific architectural modifications. However, VGG based

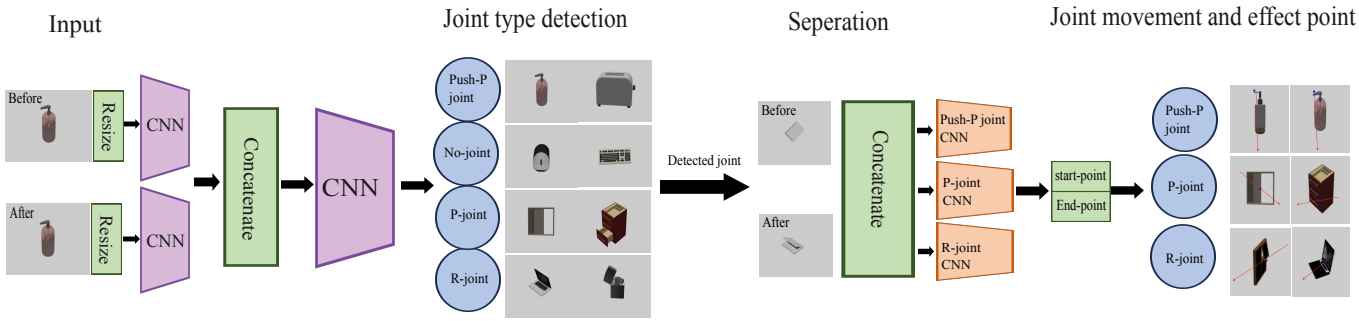

Fig. 2. JoTyMo, consists of two steps: classification and detection. A two-layer CNN to classify the joint types of the objects from RGB images before and after an interaction is used. Then, another CNN, trained on a specific joint type, to detect the movement vector and the effect point of the object is utilized. The inputs to the convolutional layers are the same images used for classification. If the object has no joints, the images are taken from different viewpoints.

layers were chosen to avoid complex structure suggested in [25].

For classification purposes, softmax layer is employed after the dense layers, as described in [27]. Additionally, in JoTyMo, Rectified Linear Units (ReLU) are applied to the activation functions in both CNN layers and fully-connected layers , inspired by the findings of [28], where using ReLU as the activation function in all neural network layers resulted in higher accuracy in classification and regression.

B. CNN-based Joint State Estimation

In [2], a general overview is provided on the Joint state estimation using CNNs. In addition, works such as [29] , [30] and [31] have developed methods based on CNNs to estimate objects joint state. In [32] CNNs are utilized in order to estimate pose which is similar to movement detection. In order to enhance the accuracy of joint detection, [32] introduced a localization approach that involves identifying and separating the joints. The method utilizes a CNN to initially localize the joints as 2D keypoints. Furthermore, self-attention is employed between the CNN features at these key points to establish associations between the key points and their respective hand joints. This strategy, evident in [32], effectively improves the detection task. In this paper, following the same reasoning as the one presented in [32], the proposed model, is designed based on CNNs.

C. CNN encoder architecture

The proposed technique in [1] is centered around object tracking, relying on an object tracking method that utilizes two CNNs trained offline [1]. The main idea is to alternate between tracking the object using motion information and predicting its temporal location based on visual similarity [1]. Similarly, [24] adopts a similar approach by employing two CNN encoders. The second encoder takes the extracted features from the initial CNN and applies them to input images. The two parallel layers of feature extraction and concatenation, followed by feeding the data to a second level of CNN, depicted in Fig. 2, were inspired by the works of [1] and [24].

D. Articulated Object Modeling

The works proposed in [3] and [26] address the challenge of detecting objects with openable parts within a single image. Furthermore, in [26], the authors extend the openable part detection task to encompass scenes with multiple objects, and they develop a corresponding dataset based on real-world scenes. Additionally, [33] explores modeling indoor scenes through interactive perception. Similarly, the paper [34] employs interaction to model articulated objects in both simulated and real environments.

JoTyMo expands upon the scope of [3] by incorporating RGB images of simulated articulated objects following interaction. This approach acknowledges the importance of interaction for accurate detection, mirroring the suggested method in [34].

III. Joint Type and Movement Detection in Articulated Objects Using CNNs

The proposed method introduced in this section aims to determine whether an object possesses joints, identify the specific type of joint, understand its range of motion, and pinpoint the possible grasping point for effective joint manipulation in [6]. This probable grasp point is commonly referred to as the ”effect point”. Moreover, it is crucial to gather data regarding the trajectory of the joint’s movement to accurately model the objects. This is why it is necessary to identify the endpoint of movement, which signifies the need for pre and post-interaction images. In addition, it is imperative to ascertain the joint type beforehand due to the variations in joint limitations. Therefore, a classification network is implemented to precede the regression network for the articulated object.

In this section, firstly, the data set that was used and the reasons behind choosing it over utilizing other similar datasets is presented. Additionally, a comparison is made between these different data sets, and the rationale for selecting PartNet-Mobility is explained. Moreover, the proposed augmentation is discussed, which aims to generate unseen images. Following that, the architecture employed to accomplish the task is

TABLE I
DIFFERENT ARTICULATED OBJECT DATA SETS COMPARISON.

Database name	Database features		
	Data type	*Number of Categories*	*Data Environment*
Shape2Motion [13]	Point Cloud	10	Real and Simulated
ShapeNet-Core [14]	3D CAD models	3135	Simulated
SAPIEN PartNet-Mobility [6]	**RGB**	**2037**	**Simulated**

described, and the constituent elements within it are explained. Within this section, a method is presented for regularization in convolutional layers. Furthermore, for the sake of enhanced clarity, a summarized depiction of the design is provided in Fig. 2.

A. Choosing the most compatible Articulated Objects Data set

There are multiple datasets available that feature articulated objects, as shown in table I. Shape2motion [13], offers both simulated and real articulated objects. However, due to its use of point cloud data, utilizing this dataset requires significant computational effort. [13] also contains only 10 categories of objects in comparison to the other datasets mentioned in Table I. Another dataset, ShapeNet-Core [14], offers the advantage of including a wide variety of categories and objects based on CAD. However, it falls short in one aspect as it does not provide Push-P joints. On the other hand, PartNet-Mobility dataset does include Push-P joints. This feature allows for the detection of four different categories of interactive objects, which is an improvement compared to the other two datasets that only have three categories. In summary, PartNet-Mobility encompasses four distinct categories of objects in terms of interaction. These include P-joints, which are characterized by drawers, sliding doors, and cutters. Additionally, R-joints, such as laptops, lighters, and cabinet doors, present their own set of complexities. The third category, PushP-joints, comprises dispensers, toasters, and similar objects. Lastly, there are non-articulated objects like keyboards, mice, remote controls, and cameras. Collectively, these categories provide a comprehensive range of intractable objects within the PartNet-Mobility framework. Moreover, the importance of using different view angles and significance of having Push-P joints means that using [6] is the only viable option as the two other do not include such a joint type, while the other two data bases provide either more groups of data similar to [14] or a range of simulated and real world options similar to [13]. In addition, figure 1 showcases some examples of objects that exist in [6].

Overall, a total of 200 images were captured from 4 different joint types across 40 different objects. Each image had dimensions of $224 \times 224 \times 3$ and was selected from a database either before or after an interaction. In order to establish a comprehensive dataset, 600 training images and 120 validation images were derived from the initial selection of 200 baseline images. Each image in the dataset encompassed two versions depicting the before and after interaction, with dimensions of $224 \times 224 \times 3$.

In order to optimize the efficiency and alignment of the utilized neural networks, augmentation techniques were employed. Specifically, horizontal and vertical flips were chosen as augmentations, along with scale and intensity adjustments. These specific augmentations were carefully selected to ensure smooth training and validation processes, devoid of any potential complexities. Figure 3 illustrates a variety of examples of these augmentations, alongside the corresponding joint type identified by JoTyMo.

B. Network architecture and network setting used in articulated object detection and modeling

Two RGB images, resized to $96 \times 96 \times 3$ dimensions, are derived from the original images before and after processing. These images are then inputted into a convolutional layer, as depicted in Fig. 2. The resulting shape of each smaller convolutional layer is $21 \times 21 \times 96$. Subsequently, these two outputs are concatenated to produce a $21 \times 21 \times 192$ vector, which is in turn passed into another CNN for classification purposes.

The applied loss function is the Categorical Cross Entropy, enabling the generation of output probabilities for each grouped category. The identified joint type is determined by selecting the highest probability from these probabilities. To summarize, the main goal of the initial segment of the architecture seen in Fig. 2 is to obtain a transformation function from two input images to the class probability.

A similar transform is used for finding the joint movement vector and effect point, based on the fact that joint type is known. The Latter means that each joint type has its own dedicated trained Network. The primary distinction lies in utilizing a singular path for regression. This approach is employed due to both input images being of dimensions

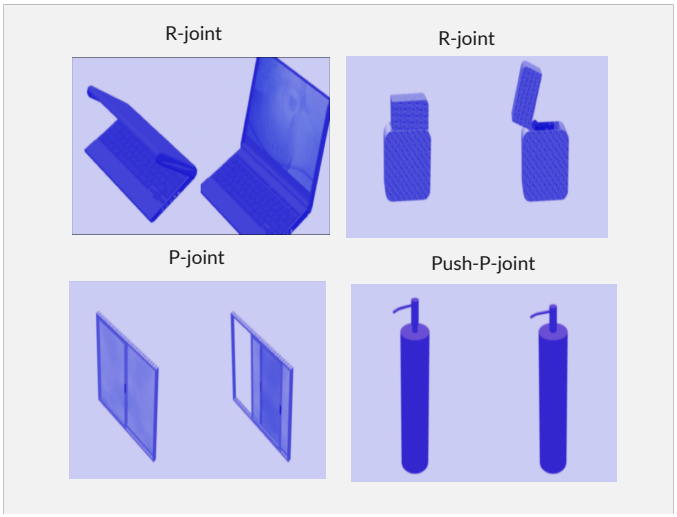

Fig. 3. This diagram highlights the application of various augmentations on the original images. Augmentations employed include intensity and color scaling, image zooming, and vertical flipping, as depicted in the figure provided. These transformations aim to enhance the overall variation of the images in both training and testing in order to increase the generalization ability of JoTyMo.

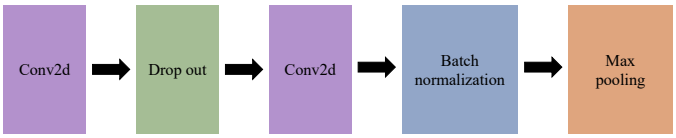

Fig. 4. This diagram illustrates the sequence of internal layers employed within convolutional layers. It begins with a conv2d operation, proceeds with a dropout layer, and is then followed by another conv2d operation. Lastly, there is a batch normalization step and a maxpooling operation.

$224 \times 224 \times 3$, without any resizing involved. By concatenating these two images and passing them through the CNN, the resultant output shape is $3 \times 3 \times 32$. After the CNN layer, two dense layers are used to transform the flattened input into 4 regression points.

Within each Convolutional layer, a design inspired by VGG-16 was implemented. This involved the utilization of a conv2d layer, followed by a dropout layer, and then another conv2d layer. Subsequently, batch normalization was applied, followed by max pooling. The conv2d layers employed kernel sizes of 3×3, while the pool size in the maxpooling layers was 2×2. Figure 4 depicts a visual representation of this design.The loss function used in regression is referred to as "Huber Loss." It was chosen because it provides a smoother learning curve compared to other loss functions like "Mean Square Error Loss (MSE Loss)". However, for evaluation purposes, MSE Loss is used because it allows for better interpretations. Finally, utilization of L1 and L2 regularizations in the initial and final layers was applied. By incorporating this regularization techniques, the learning process was expedited, resulting in a more pronounced reduction in the loss functions.

IV. Results and Discussion

This section presents a display of the outcomes accomplished by JoTyMo in detecting the types and movements of articulated objects. An exhaustive evaluation is conducted to compare JoTyMo with classical methods and deep learning approaches, emphasizing the performance and architectural significance of the proposed approach in this paper. Moreover, the generalization ability of JoTyMo is assessed by evaluating its performance on augmented and previously unseen inputs. Lastly, a thorough examination is conducted to explore the limitations and challenges encountered by the system.

A. Training Curves

Fig. 5 shows the training and testing losses during the learning process for both the classification and regression. As it can be observed from Fig. 5, the losses of both the validation and training data decreased significantly and eventually reached a plateau across all sub-figures. Additionally, it is worth noting that the curves representing the validation and training data displayed a smooth progression. Moreover, all Huber loss values from the testing phase demonstrated a significant decrease, falling below 0.1. This indicates successful prevention of overfitting and further highlights the model's capability to achieve a satisfactory level of accuracy when predicting unseen data.The aforementioned pattern is also observable in Categorical cross entropy, wherein it exhibited a considerable value below 0.2.

TABLE II
COMPARING CLASSIFICATION METHODS WITH JOTYMO.[a]

Method	Classification Metrics[b]		
	Precision-Avg(%)	*f1-Score-Avg(%)*	*Recall-Avg(%)*
LDA(LSQR) [16]	87	86	86
SVM(Poly) [17]	92	92	92
MLP(dept=6) [19]	84	82	83
Random Forests(dept=3) [19]	82	81	82
Convolutional Neural Network	**96**	**95**	**95**

[a] Comparison done on same testing data, all models are fitted on the same training data.
[b]The bigger the percentage, the better.

B. Performance against Classical Methodology

In analysis against classical methods, two comparisons are made, namely, the effectiveness of the classification and regression tasks. Table II represents the accuracy and other metrics concerning the classifier, while Table III displays the performance of the regression task through two different metrics: the mean absolute error percentile (**MAEP**) and the mean squared error percentage (**MSEP**). These metrics provides a quantitative measure to assess the accuracy of predictions. By calculating the MAEP and MSEP, evaluation of the performance of JoTyMo can be done.

The results in Table II demonstrate the capability of JoTyMo's architecture to discern joint types by considering the observed interaction, rather than simply selecting features from the input. Moreover, as illustrated in Table III, the regression results highlight the CNN's ability to accurately estimate the movement vector, both in terms of magnitude and direction. This is evidenced by the higher MAEP and smaller angle

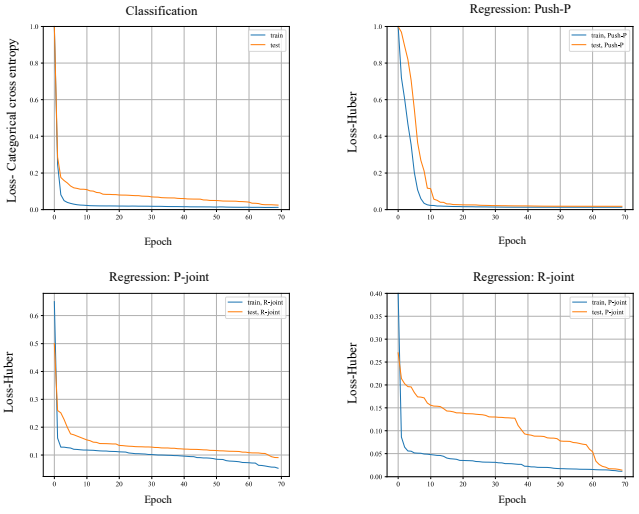

Fig. 5. The top left curve represents the loss function employed for joint detection, whereas the top right graph demonstrates the loss function specifically designed for the detection of Push-P joint movement vectors. Furthermore, the two bottom-line graphs depict the training curves for the loss function utilized in detecting P-joint and R-joint respectively.

TABLE III
COMPARING REGRESSION METHODS WITH JOTYMO[a].

Methods\Metrics	Joint Type							
	P-joint		*Push-P-joint*		*R-joint*		*Total*	
	MAEP(%)	*MSEP(%)*	*MAEP(%)*	*MSEP(%)*	*MAEP(%)*	*MSEP(%)*	*MAEP(%)*	*MSEP(%)*
SVM[b] Regression [17]	69	76	85	87	79	81	77.66	81.33
MLP Regression [19]	73	80	71	74	68	71	70.66	75.00
Random Forest Regression [19]	95	96	90	91	85	87	90.00	91.33
Voting Regression (linear + gradient boosting) [20]	73	81	85	87	58	62	72.00	76.67
Convolutional Neural Network	**92**	**95**	**91**	**91**	**99**	**96**	**94.00**	**94.00**

[a] Comparison done on same test data, all models are fitted on the same training data.
[b] With polynomial kernel

difference, as indicated by the higher MSEP. Additionally, as shown in Table III, the CNN surpasses most classical methods in performance for each joint type. Although Random Forest Regression also yields acceptable results in certain cases, it is worth noting that MLP achieves notable accuracy. However, MLP is more complex than Random Forest and less precise compared to the CNN or Random Forest approaches.

C. Comparison of Classifier CNN to Alternative Deep Learning Methods

The impact of architecture on the proposed design's performance was examined by assessing various pre-trained networks through the employment of transfer learning [18]. A single input, representing the state before and after the interaction, was provided to these networks in the form of concatenated images. In order to ensure fairness in the comparison, these networks were trained for the same duration as JoTyMo, utilizing identical loss functions and optimization algorithms. By adopting this approach, the influence of hyperparameters on the outcomes was minimized. The accuracy of both training and testing data, as showcased in Table IV, denotes the models' capacity to learn and generalize.

Different categories of joints are distinguished by JoTyMo, more accurately, as indicated by Table IV, which contrasts with other networks. Overfitting is observed in ResNet and Inception, while the VGG-based designs display underfitting. Despite having fewer parameters compared to all the transfer learned networks, suggested classifier achieves a more accurate results owing to its structure.

TABLE IV
COMPARING TRANSFER LEARNING NETWORKS WITH JOTYMO.

Network Name	Number of parameters	Accuracy Metrics	
		Accuracy-test(%)	*Accuracy-train(%)*
VGG-16 [8]	50.3M	19.00	25.73
VGG-19 [8]	55.7M	19.00	25.87
Resnet50 [21]	23.6M	28.00	93.60
Resnet50-V2 [22]	23.5M	41.00	98.62
Inception-V3 [23]	21.8M	6.00	98.84
JoTyMo	**1.2M**	**94.00**	**99.85**

D. Accuracy on Augmented Images

The performance of JoTyMo on unseen data was assessed through the pre-training of the CNN using augmented images which varied in parameters such as intensity and scale. The accuracy of design on the validation data was subsequently tested, and the results are provided in Table V. Observing Table V, it is evident that, due to the system's reliance on interaction detection, complete separation between objects without joint and articulated objects has been achieved. Furthermore, R-joints exhibit distinct movement patterns compared to P-joints and Push-P joints, allowing for a more accurate classification. On the other hand, due to the similarity in movement between P-joints and Push-P joints, their categorization is not as precise as that of the former two.

E. Limitations

JoTyMo was trained on a simulated environment, and it is expected that it will require some fine-tuning and increasing the number of filters in each layer of the network to produce reliable results in reality. Moreover, the interaction with the object should be visible in the RGB image. Otherwise, the system will classify it as an object without joints. In addition, some challenges were encountered during the testing phase, such as Push-P joints that were opened being detected as P-joints, and P-joints which were closing being occasionally predicted as Push-P joints. Figure 6 illustrates some of these examples. The Push-P joint dispenser was identified as a P-joint object. Additionally, in cases where the interaction was not observed in the picture, the laptop was deemed a jointless object, but if the interaction was visible, it was then identified as an R-joint by JoTyMo.

TABLE V
RESULTS ON ALL UNSEEN CATEGORIES.

Joint Type\ Metric	*Precision on unseen data(%)*	f1-score on unseen data(%)
No-joint	100	22
P-joint	50	67
Push-P-joint	100	77
R-joint	86	92
Maco-Avg	**84**	**65**

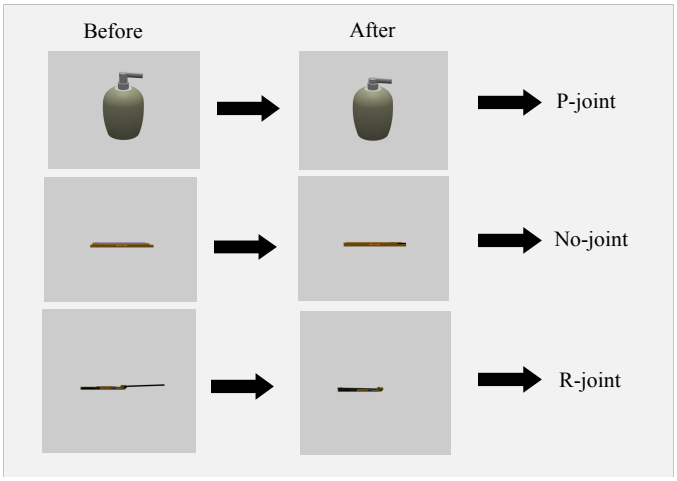

Fig. 6. The figure demonstrates how the algorithm detects different types of joints in the images. In row 1, the algorithm misclassified a Push-P joint as a P-joint, because it did not capture the opening between the two objects. In rows 2 and 3, the algorithm correctly identified a R-joint, but failed to recognize the interaction between the objects in row 2, and labeled it as a single object without a joint.

V. Conclusion

This paper introduced a CNN-based architecture that addressed the task of detecting the type of joint present in an articulated object and analyzing its potential movement. By leveraging two images, one captured before the interaction and another taken after, this architecture surpasses classical methods and transfer learning approaches on the PartNet-Mobility dataset. What is more, it achieved an accuracy rate of 96% for joint type detection and 94% for joint movement detection. These results underscored the effectiveness and performance of the proposed architecture.The proposed model showcased performance on unseen data, highlighting its impressive generalization capability, particularly in accurately differentiating between objects without joints and those with joints, achieving near-perfect accuracy.

JoTyMo can also serve as an effective and fast preprocessor for a larger system which aims at learning how to grasp articulated objects through interactive learning. Moreover, JoTyMo offers an approach to classify and detect objects which have visible moving parts based on their motion rather than their type. However, JoTyMo is not a complete solution and there are many ways to enhance and modify it. For instance, a multimodal JoTyMo can be designed that utilizes text or audio inputs from humans to guide the detection process. Another possibility is to use vision transformers instead of CNNs to improve generalization and create a system that can handle the grasping task end-to-end rather than as a preprocessing step.

References

[1] R. Tapu, B. Mocanu, and T. Zaharia, "DEEP-SEE: Joint Object Detection, Tracking and Recognition with Application to Visually Impaired Navigational Assistance," Sensors, vol. 17, no. 11, p. 2473, Oct. 2017, doi: 10.3390/s17112473.

[2] K. Młodzikowski and D. Belter, "CNN-based Joint State Estimation During Robotic Interaction with Articulated Objects," 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, Singapore, 2022, pp. 78-83, doi: 10.1109/ICARCV57592.2022.10004277.

[3] Jiang, Hanxiao, Yongsen Mao, Manolis Savva and Angel X. Chang. "OPD: Single-view 3D Openable Part Detection." European Conference on Computer Vision (2022), in press.

[4] Alzubaidi, L., Zhang, J., Humaidi, A.J. et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8.

[5] Aman Shrivastav, "Different types of CNN models" OpenGenus. https://iq.opengenus.org/different-types-of-cnn-models(Auguest 21th, 2023).

[6] Xiang, Fanbo, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, et al. 2020. "SAPIEN: A SimulAted Part-Based Interactive ENvironment." ArXiv.org. March 18, 2020. https://doi.org/10.48550/arXiv.2003.08515.

[7] A. Krizhevsky, "CIFAR-10 and CIFAR-100 datasets," Toronto.edu, 2009. https://www.cs.toronto.edu/ kriz/cifar.html

[8] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv.org, Apr. 10, 2015. https://arxiv.org/abs/1409.1556.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, May 2012, doi: https://doi.org/10.1145/3065386.

[10] F. Siahkali, S. A. Alavi and M. T. Masouleh, "SIVD: Dataset of Iranian Vehicles for Real-Time Multi-Camera Video Tracking and Recognition," 2022 8th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), Behshahr, Iran, Islamic Republic of, 2022, pp. 1-7, doi: 10.1109/ICSPIS56952.2022.10043932.

[11] H. Jalali, S. Samadi, A. Kalhor and M. T. Masouleh, "Model-Free Dynamic Control of a 3-DoF Delta Parallel Robot for Pick-and-Place Application based on Deep Reinforcement Learning," 2022 10th RSI International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran, Islamic Republic of, 2022, pp. 48-54, doi: 10.1109/ICRoM57054.2022.10025269.

[12] K. Morabia, J. Arora, and T. Vijaykumar, "Attention-based Joint Detection of Object and Semantic Part," arXiv.org, Jul. 05, 2020. https://arxiv.org/abs/2007.02419 (accessed Aug. 26, 2023).

[13] X. Wang, B. Zhou, Y. Shi, X. Chen, Q. Zhao, and K. Xu, "Shape2Motion: Joint Analysis of Motion Parts and Attributes from 3D Shapes," arXiv.org, Mar. 12, 2019. https://arxiv.org/abs/1903.03911 (accessed Aug. 26, 2023).

[14] A. X. Chang et al., "ShapeNet: An Information-Rich 3D Model Repository," arXiv:1512.03012 [cs], Dec. 2015, Available: https://arxiv.org/abs/1512.03012.

[15] Kingma, Diederik P, and Jimmy Ba. "Adam: A Method for Stochastic Optimization." ArXiv.org, 22 Dec. 2014, arxiv.org/abs/1412.6980.

[16] Ye, Jieping. (2007). Least squares linear discriminant analysis. Proceedings of the 24th international conference on Machine learning. 227. 1087-1093. 10.1145/1273496.1273633.

[17] Vinge, Rikard & Mckelvey, Tomas. (2019). Understanding Support Vector Machines with Polynomial Kernels. 1-5. 10.23919/EUSIPCO.2019.8903042.

[18] Hosna, Asmaul, et al. "Transfer Learning: A Friendly Introduction." Journal of Big Data, vol. 9, no. 1, 22 Oct. 2022, https://doi.org/10.1186/s40537-022-00652-w.

[19] Trappenberg, Thomas. (2019). Machine learning with sklearn. 10.1093/oso/9780198828044.003.0003.

[20] Erdebilli, Babek & Devrim-İçtenbaş, Burcu. (2022). Ensemble Voting Regression Based on Machine Learning for Predicting Medical Waste: A Case from Turkey. Mathematics. 10. 2466. 10.3390/math10142466.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," arXiv.org, Dec. 10, 2015. https://arxiv.org/abs/1512.03385.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," arXiv:1603.05027 [cs], Jul. 2016, Available: https://arxiv.org/abs/1603.05027.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," arXiv.org, 2015. https://arxiv.org/abs/1512.00567.

[24] M. Seeland and P. Mäder, "Multi-view classification with convolutional neural networks," PLOS ONE, vol. 16, no. 1, p. e0245230, Jan. 2021, doi: https://doi.org/10.1371/journal.pone.0245230.
[25] V. Santhanam, V. I. Morariu, and L. S. Davis, "Generalized Deep Image to Image Regression," arXiv.org, Dec. 10, 2016. https://arxiv.org/abs/1612.03268 (accessed Sep. 02, 2023).
[26] [9]X. Sun, H. Jiang, M. Savva, and A. X. Chang, "OPDMulti: Openable Part Detection for Multiple Objects," arXiv.org, Mar. 24, 2023. https://arxiv.org/abs/2303.14087.
[27] A. A. Mohammed and V. Umaashankar, "Effectiveness of Hierarchical Softmax in Large Scale Classification Tasks," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Sep. 2018, doi: https://doi.org/10.1109/icacci.2018.8554637.
[28] Agarap, Abien Fred, "Deep Learning using Rectified Linear Units (ReLU)," arXiv.org, 2018. https://arxiv.org/abs/1803.08375
[29] X. Fang and X. Lei, "Hand pose estimation on hybrid CNN-AE model," IEEE Xplore, Jul. 01, 2017. https://ieeexplore.ieee.org/abstract/document/8079051.
[30] Y. Kim and D. Kim, "A CNN-based 3D human pose estimation based on projection of depth and ridge data," Pattern Recognition, vol. 106, p. 107462, Oct. 2020, doi: https://doi.org/10.1016/j.patcog.2020.107462.
[31] L. Ding, Y. Wang, R. Laganière, D. Huang, and S. Fu, "A CNN model for real time hand pose estimation," Journal of Visual Communication and Image Representation, vol. 79, p. 103200, Aug. 2021, doi: https://doi.org/10.1016/j.jvcir.2021.103200.
[32] S. Hampali, S. D. Sarkar, M. Rad, and V. Lepetit, "Keypoint Transformer: Solving Joint Identification in Challenging Hands and Object Interactions for Accurate 3D Pose Estimation," openaccess.thecvf.com, 2022, in press.
[33] Z. Jiang, C.-C. Hsu, and Y. Zhu, "Ditto: Building Digital Twins of Articulated Objects from Interaction," arXiv:2202.08227 [cs], Mar. 2022, Available: https://arxiv.org/abs/2202.08227.
[34] Z. Jiang, C.-C. Hsu, and Y. Zhu, "Ditto: Building Digital Twins of Articulated Objects from Interaction," arXiv:2202.08227 [cs], Mar. 2022, Available: https://arxiv.org/abs/2202.08227.
[35] [3]"Traffic Signs Recognition using CNN and Keras," ieeexplore.ieee.org. https://ieeexplore.ieee.org/document/10100276 (accessed Sep. 28, 2023).