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 introduces JoTyMo, a system designed to classify and detect joint types and movements of articulated objects within the SAPIEN PartNet-Mobility Data set. The approach utilizes a two-stage process, employing consecutive CNN encoders based on the VGG architecture to classify joints from pre and post-interaction images. Additionally, another Convolutional layer is utilized to regress both an effect point and movement vector, based on the joint type provided.Moreover, extensive evaluation showcases notable results, achieving 96% accuracy in joint classification and 94% accuracy in regression on the dataset. Furthermore, a comprehensive comparison is provided with existing methods, highlighting JoTyMo's performance and its ability to effectively generalize on simulated objects.**

I. INTRODUCTION

Grasping is one of the fundamental but most challenging skills of robots. Grasping is also important for enabling robotics arms to perform various tasks in various environments. One of the main challenges of robotic grasping is how to deal with objects that have restrictions in movement such as articulated objects. Using the convolutional neural networks(CNNs) in articulated object grasping has been 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. Moreover, various applications, such as robotic manipulation, object recognition, and human-computer interaction calls for processing more than one image per object. There are also works such as Multi-Camera Video Tracking and Recognition [10], that utilize CNN to track objects and classify them, promote the usage of CNN in classifying images.

There are different approaches to tackle this problem, 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 learning-based methods, which use deep neural networks to learn the joint state from data have also been suggested.

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 can allow them to adapt to changing or uncertain situations, and to handle complex or nonlinear dynamics that might be difficult to model [4].

They can also leverage existing techniques and frameworks, such as such as convolutional neural networks (CNNs), which have been proven to be effective in various tasks like image classification, object detection, and natural language processing. By utilizing CNNs, 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 scale to large datasets and handle high-dimensional data efficiently. [5]. Subsequently, Methods such as DEEP-SEE [1] ,CNN-based Joint State Estimation [2] and Single-view 3D Open-able Part Detection [3] have been suggested.

This paper introduces a CNN-based approach for the identification of joint types and movements of articulated objects using RGB images. The chosen dataset for training and testing the proposed method is the PartNet-Mobility Dataset [6], which comprises images captured before and after interactions or view changes. The objects are classified into four joint types: P-joint, R-joint, Push-P joint, and Not articulated. Figure 1 showcases several examples of these images.

Furthermore, an additional CNN architecture is employed to train on a specific joint type. This facilitates the estimation of the movement vector and the joint effect point of the object post-classification. To assess its accuracy and design, the obtained results from the JoTyMo model are compared with various classical machine learning algorithms and transfer learning techniques. Lastly, a review of related works is conducted before concluding this section.

A. Deep Convolutional Networks for Image Classification and Regression

Various architectures have been proposed for image classification, including VGG [20], Alexnet [9], Inception-V3

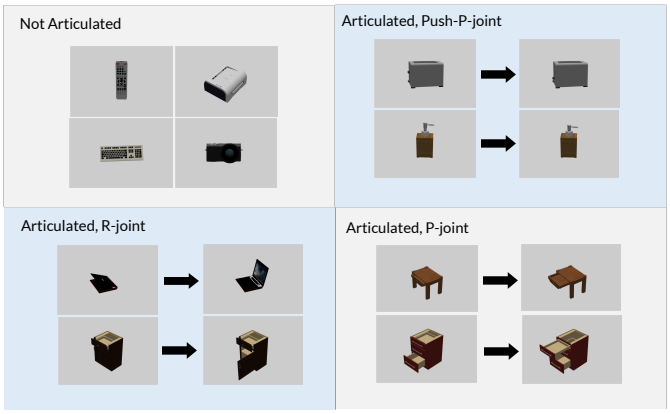

Fig. 1. This visual representation showcases various examples of distinct objects along with their potential interactions within the PartNet-Mobility Dataset [6].

[23], and ResNets [22], [23]. Among these, the VGG-based design was selected due to its use of smaller kernel sizes. To address image-to-image regression, a method was developed in [25]. Furthermore, [26] introduces an active learning approach. However, VGG was chosen to avoid more complex structures. For classification, a softmax layer is employed after the dense layers, as described in [28]. Additionally, Rectified Linear Units (ReLU) are utilized as the activation functions, following the findings of [29].

B. CNN-based Joint State Estimation

The paper [2], provides a general overview on the Joint state estimation using convolutional neural networks (CNNs). In addition, works such as [30] , [31] and [32] have developed methods based on CNNs to estimate an objects joint state. [33] also utilized CNNs to estimate pose, which is similar to movement detection. Because of that, JoTyMo also was designed based on CNNs instead of transformer design suggested in [34].

C. Using Multiple CNN Networks

The fundamental essence of the suggested technique in [1] relies on an object tracking method, which in turn relies on the utilization of two convolutional neural networks that have been trained offline [1]. The primary principle entails the alternation between tracking through the utilization of motion information and predicting the temporal location of the object based on visual similarity [1]. The paper [24] also utilizes a similar approach of using two layers of neural networks where the second one uses data from the first layer that has been applied to input images.

D. Articulated object Modeling

The works proposed in [3] and [27] address the challenge of detecting objects with openable parts within a single image. Furthermore, in [27], 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, [36] explores modeling indoor scenes through interactive perception. Similarly, the paper [37] employs interaction to model articulated objects in both simulated and real environments.

This paper 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 [37].

II. Methodology

The goal is to classify RGB images that are generated based on interaction in the simulation environment of [6]. Moreover, it is aimed to determine the location of the effect point and movement vector of the mentioned interaction, taking into consideration the known joint type. Furthermore, it is assumed that the presence of joint interaction can be observed in the RGB image, if it exists.

Firstly, the introduction formally presents the data set used and the task at hand. Different data sets are compared and the rationale for selecting Part-net Mobility is explained. Additionally, the augmentation performed to generate unseen images is proposed. Then, the architecture employed to accomplish the task is described and the constituent elements within it are explained. In this subsection, a method is presented that resulted in visually seamless training. . Furthermore, for enhanced clarity, a summarized depiction of the design is provided in figure 2.

A. Data set

There are several data sets available that contain articulated objects. These data bases are presented in table I. In addition, figure 1 showcases some examples of objects that exist in the data set [6]. 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 [14] or a range of simulated and real world options [13].

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.

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 $224 \times 224 \times 3$.

TABLE I
DIFFERENT DATA SETS COMPARISON

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

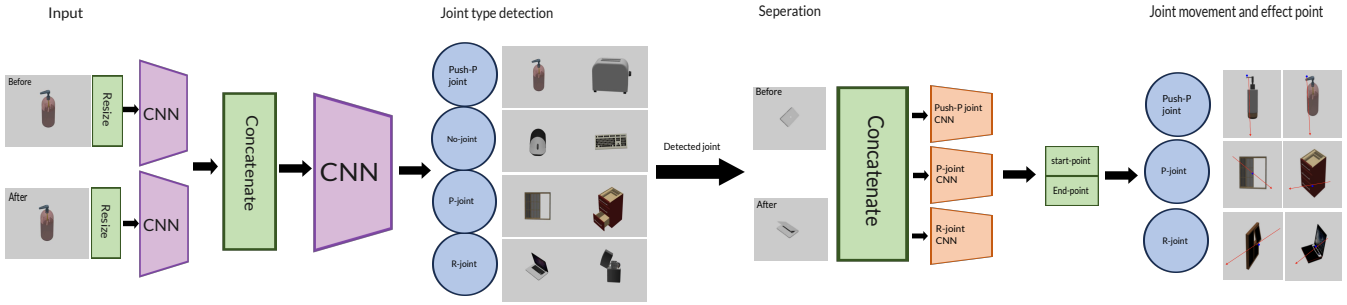

Fig. 2. JoTyMo, consists of two steps: classification and detection.A two-layer CNN to classify the joint types of the objects from RGB images before and after an interaction is used. Then, another CNN, trained on a specific joint type, to detect the movement vector and the effect point of the object 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.

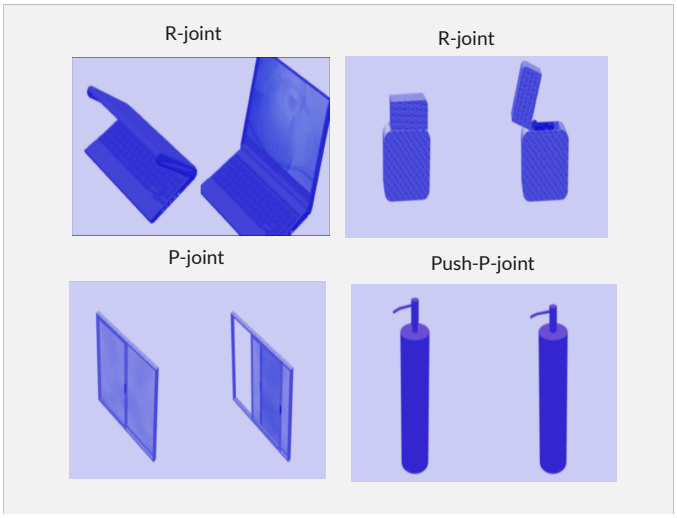

Fig. 3. This figure illustrates how the original images are transformed by applying different types of augmentations, such as changing the color intensity and scaling the size.

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 showcases a variety of examples of these augmentations, alongside the corresponding joint type identified by JoTyMo.

B. Problem Formulation and Proposed solution

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, depicted in Figure 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 implemented loss function is the Categorical Cross Entropy, which allows for the generation of output probabilities for each grouped category. The detected joint type is determined by selecting the highest probability within these probabilities.

In summary, the primary objective of the initial portion of the architecture is to acquire 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. What it means is that each joint type has it's own dedicated trained Network. The only difference here is that instead of having two paths, only one path is used for the regression because the input both images are $224 \times 224 \times 3$ and no resize is used. The output of the CNN is $3 \times 3 \times 32$. After the CNN, 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.

III. Results and Discussion

This section presents a display of the outcomes accomplished by JoTyMo in detecting the types and movements

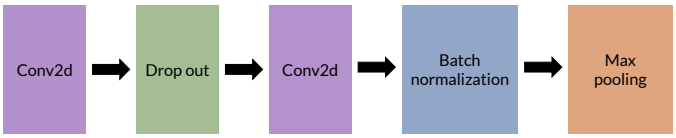

Fig. 4. This diagram illustrates the sequence of internal layers employed within convolutional layers. It begins with a conv2d operation, proceeds with a dropout layer, and is then followed by another conv2d operation. Lastly, there is a batch normalization step and a maxpooling operation.

of articulated objects. An exhaustive evaluation is conducted to compare JoTyMo with classical methods and deep learning approaches, emphasizing the superior performance and architectural significance of our system. 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

Figure 5 shows the training and testing losses during the learning process for both the classification and regression.

In Figure 5, it can be observed that 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.

B. Performance against Classical methodology

In this analysis, two comparisons are to be made: the effectiveness of the classification and regression tasks. Table II presents the accuracy and other metrics concerning the classifier, while Table IV showcases 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 superior capability of our architecture to discern joint types by considering the observed interaction, rather than simply selecting features from the input. Moreover, 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 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.

TABLE II
COMPARING CLASSIFICATION METHODS ON THE SAME TRAINING DATA[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] Done on Validation data, as all models are fitted on the training perfectly.

[b]The bigger the percentage, the better.

C. Comparison of classifier CNN to alternative deep learning methods

The impact of architecture on our 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. To ensure fairness in the comparison, these networks were trained for the same duration as our proposed CNN, 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, as indicated by Table III, which contrasts with other networks. Overfitting is observed in ResNet and Inception, while the VGG-based design displays underfitting. Despite having fewer parameters compared to all the transfer learned networks, suggested classifier achieves an optimal result owing to its structure.

D. Accuracy on augmented images

The performance of JoTyMo on unseen data was assessed through the pre-training of the CNN using augmented images that varied in parameters such as intensity and scale. The accuracy of design on the validation data described in II-A 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,

TABLE III
COMPARING REGRESSION METHODS ON THE SAME TRAINING DATA[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**

[a] Done on Validation data, as all models are fitted on the training perfectly.
[b] with Polynomial kernel

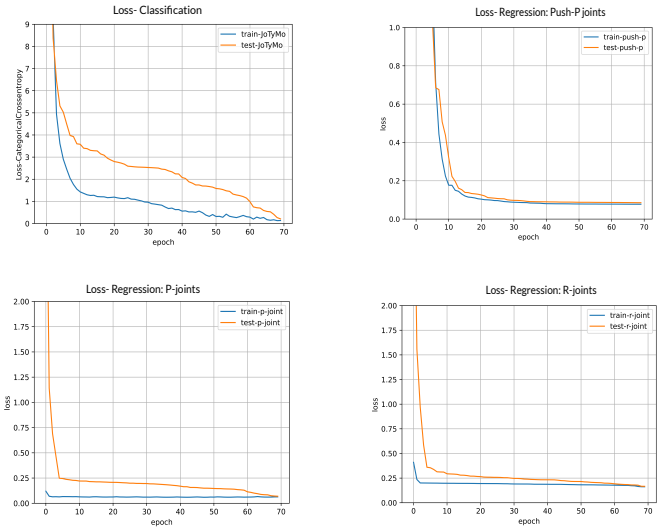

Fig. 5. The top left curve represents the loss function employed for joint detection, whereas the top right graph demonstrates the loss function specifically designed for the detection of Push-P joint movement vectors. Furthermore, the two bottom-line graphs depict the training curves for the loss function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

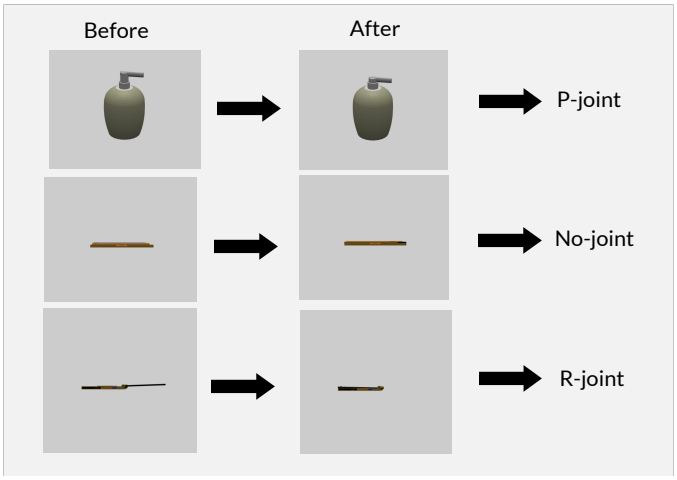

Fig. 6. The figure demonstrates how the algorithm detects different types of joints in the images. In row 1, the algorithm misclassified a push-p joint as a p-joint, because it did not capture the opening between the two objects. In rows 2 and 3, the algorithm correctly identified a R-joint, but failed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

TABLE IV
COMPARING NETWORKS

Network Name	Number of parameters	Accuracy Metrics	
		Accuracy-test(%)	*Accuracy-train(%)*
VGG-16 [8]	50.3M	25.73	19
VGG-19 [8]	55.7M	25.87	19
Resnet50 [21]	23.6M	28	93.6
Resnet50-V2 [22]	23.5M	41	98.62
Inception-V3 [23]	21.8M	6	98.84
JoTyMo	**1.2M**	**94**	**99.85**

and p-joints that were closing being occasionally predicted as push-p joints. Figure 6 illustrates some of these examples.

IV. CONCLUSION

In conclusion, JoTyMo was developed as a system that can achieve higher accuracy in detecting joint types and movements of articulated objects than existing methods. The suggested model also performed well on unseen images,

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**

demonstrating its generalization ability. JoTyMo can also serve as an effective and fast preprocessor for a larger system that aims to learn how to grasp articulated objects through interactive learning. Moreover, JoTyMo offers an approach to classify and detect objects that 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 in [35] and [38] 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] Ranganathan, H., Venkateswara, H., Chakraborty, S., Panchanathan, S. (2020). Deep Active Learning for Image Regression. In: Wani, M., Kantardzic, M., Sayed-Mouchaweh, M. (eds) Deep Learning Applications. Advances in Intelligent Systems and Computing, vol 1098. Springer, Singapore. https://doi.org/10.1007/978-981-15-1816-4_7.

[27] [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 (accessed Sep. 03, 2023).

[28] 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.

[29] Agarap, Abien Fred, "Deep Learning using Rectified Linear Units (ReLU)," arXiv.org, 2018. https://arxiv.org/abs/1803.08375

[30] 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 (accessed Sep. 20, 2023).

[31] 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.

[32] 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.

[33] 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.

[34] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models," arXiv.org, Jul. 12, 2023. https://arxiv.org/abs/2307.05973.

[35] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," arXiv:2010.11929 [cs], Oct. 2020.

[36] 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.

[37] 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.

[38] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," arXiv:2103.14030 [cs], Mar. 2021, Available: https://arxiv.org/abs/2103.14030