SHARED TRANSFORMER ENCODER WITH MASK-BASED 3D MODEL ESTIMATION FOR CONTAINER MASS ESTIMATION

Tomoya Matsubara, Seitaro Otsuki,* Yuiga Wada,* Haruka Matsuo, Takumi Komatsu, Yui Iioka, Komei Sugiura and Hideo Saito*

Keio University, Japan
tomoya.matsubara@hvrl.ics.keio.ac.jp

ABSTRACT

For human-safe robot control in human-to-robot handover, the physical properties of containers and fillings should be accurately estimated. In this paper, we propose a Transformer encoder that shares the same architecture and parameters for filling level and type estimation. We also propose a mask-based geometric algorithm to estimate 3D models of containers for the estimation of their capacity and dimensions. We further use these estimations to estimate their mass in a Convolutional Neural Network model. Experiments show that our Transformer model produced encouraging results in both estimations. While challenges remain in our mask-based algorithm and Convolutional Neural Network model, their results revealed several ways for improvement.

***Index Terms*—** Transformer encoder, visual hull, Mask R-CNN, point cloud

1. INTRODUCTION

Accurate estimations of physical properties (e.g., mass and dimensions) are essential in human-to-robot handovers of objects [1, 2, 3]. As robot control often depends on their physical properties, inaccurate estimations can cause unexpected behavior and put users in danger [4]. Under limited prior data and knowledge, however, devising a robust method that can make accurate estimations even for unseen objects is never easy due to the variety of configurations [2, 5].

RGB or RGB-D images of pouring scenes are used in existing methods for the reconstruction of containers' 3D models and their dimensions estimations [6, 7]. However, these methods are only available for opaque containers whose 3D models are known. Learning the Grasping Point [8] can be used without 3D models and for transparent containers but only estimate their localization in 3D, not their dimensions.

As for fillings in containers, filling levels are estimated from RGB or RGB-D images [9, 10, 11], audio recordings [12, 13] of pouring scenes, or both [14]. Among them, a few methods estimate filling levels based on the results of filling type estimations [13], or estimate both filling levels and types in an architecture [14]. However, such architectures that combine and process both estimations for a wider variety of filling types are not yet investigated.

In this paper, we propose three methods: the first one is for simultaneous estimations of filling levels and types, the second one for the estimation of container capacities and dimensions, and the last one for container mass estimations [1]. Our models take as input audio-visual recordings of a person subject pouring a filling into a container or shaking an already-filled container.

2. FILLING LEVEL AND TYPE CLASSIFICATION

We formulate the filling level classification as a 3-class classification among *empty, half full*, and *full*, and the filling type classification as a 4-class classification among *no content, pasta, rice*, and *water*.

To tackle the two estimations, we propose a model composed of a Convolutional Neural Network (CNN) encoder, a Transformer encoder, and two classification heads (see Figure 1). We share the same architecture and parameters of the CNN and Transformer encoder for both estimations, while we use a task-specific Multi-Layer Perceptron (MLP) head in each task after the Transformer encoder. The model takes as input audio signals pre-processed and transformed into mel-spectrograms in logarithmic scale and outputs estimations of filling level and type. We consider that type estimation does not require the entire mel-spectrograms and assume even 20% of the audio could have enough information. Since the manipulation is less likely to continue from the very beginning of the recording until the very end, we discard the mel-spectrograms before time index $start$ and after time index end, which are defined by:

- $start$: randomly chosen from 0 to 40% of T
- end: randomly chosen from 60 to 100% of T

where T is the time length of the entire audio.

We use the cross-entropy loss to evaluate the training and validation loss. To avoid overfitting, we save the model only when the following condition is satisfied:

$$\mathcal{L} > \max(0.15, \mathcal{L}_l) + \max(0.15, \mathcal{L}_t) \tag{1}$$

where $\mathcal{L}$ is initialized to float('inf') of Python [15] while $\mathcal{L}_l$ and $\mathcal{L}_t$ are the validation loss of filling level and type estimation. $\mathcal{L}$ is updated, after saving the model, by $\mathcal{L} = \mathcal{L}_l + \mathcal{L}_t$. While Equation 1 is always true in the first epoch, it requires the model to make the sum of the two losses less than at least the previous sum in the following epochs. If the two losses are sufficiently small, the two constants 0.15, larger than $\mathcal{L}_l$ and $\mathcal{L}_t$, stop the training.

2.1. CNN encoder

The obtained mel-spectrograms $\boldsymbol{x}_{spec}$ have multi-channel 2D shapes (C, T, N_{mel}), where C is the number of channels and N_{mel} is the number of mel filter banks. We reshape $\boldsymbol{x}_{spec}$ using a CNN encoder composed of two layers: a depthwise convolutional layer of a kernel of size 5×5 with a padding of size 2 and a pointwise convolutional layer of a single kernel [16]. The CNN encoder combines the channels of the audio in one, so that x_{spec} can be fed to the Transformer

*Equal contribution

[1]Our source code is available on GitHub: `https://github.com/YuigaWada/CORSMAL2021`.

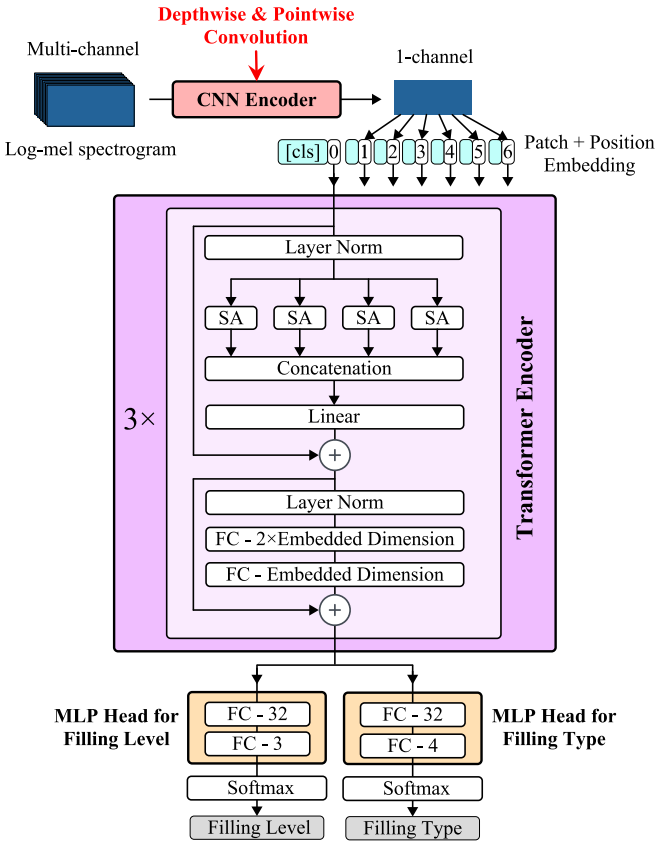

Fig. 1. Overview of the model for filling level and type classifications. The Transformer encoder takes mel-spectrogram patches as input in the training, while the estimation is computed from the first patch. The figure depicts the model during the estimation. KEY – CNN: Convolutional Neural Network, MLP: Multi-Layer Perceptron, cls: classification embedding, SA: Self-Attention, FC: Fully Connected layer.

encoder. The output of the CNN encoder $\boldsymbol{x} \in \mathbb{R}^{T \times N_{mel}}$ is thus obtained by removing the channel dimension, which has a size 1.

2.2. Transformer encoder

We prepend a learnable classification embedding $\boldsymbol{x}_{cls} \in \mathbb{R}^{N_{mel}}$ to $\boldsymbol{x}$. It serves as the aggregate representation of $\boldsymbol{x}$; its state at the output of the Transformer encoder is used to conclude the classifications. To provide $[\boldsymbol{x}_{cls}; \boldsymbol{x}]$ with information on the order of the sequence, we add sinusoidal positional encodings $\boldsymbol{P} \in \mathbb{R}^{(T+1) \times N_{mel}}$ [17]. Hence, the input of the Transformer encoder $\boldsymbol{h}$ is $\boldsymbol{h} = [\boldsymbol{x}_{cls}; \boldsymbol{x}] + \boldsymbol{P}$. The Transformer consists of three encoder blocks. Each block has a 4-head self-attention layer, two Layer Norm layers, and two Fully Connected (FC) layers. The output of the Transformer encoder is expressed as:

$$\boldsymbol{z} = [\boldsymbol{z_0}; \boldsymbol{z_1}; \cdots; \boldsymbol{z_T}] \in \mathbb{R}^{(T+1) \times N_{mel}} \tag{2}$$

where $\boldsymbol{z}_0$ corresponds to the final state of $\boldsymbol{x}_{cls}$.

$\boldsymbol{z}_0$ is fed to a classification head composed of two FC layers. For each estimation, we use a different MLP head, H_l for filling level and H_t for filling type. The output dimension of H_l is n_l, while that of H_t is n_t, where $n_l (= 3)$ and $n_t (= 4)$ represent respectively the number of classes in filling levels and types. H_l and H_t output the probabilities of each class, which are then passed into a softmax function that gives the final classification.

3. CONTAINER CAPACITY, DIMENSIONS AND MASS ESTIMATION

We consider estimations of the capacity, dimensions and mass of a container having rotational symmetry with a filling in the following setup: three fixed cameras record the manipulation of a container with a filling from an opposite view, with one view (View 1) frontal of the person subject and the other two (View 2 and 3) at a higher position at the sides of View 1. As View 2 and 3 look down at the manipulation, they provide third-person views.

To efficiently and accurately estimate container capacity and dimensions, we propose an algorithm divided into best frame selection, 3D points estimation, and capacity/dimensions estimation. It first estimates the 3D model of the container and then estimates the capacity and dimensions (top/bottom width and height) from it. As for the mass estimation, we propose a deep learning method that consists of the best frame selection, partially occluded mask restoration, and mass estimation. For mass estimation, we only use video frames from View 1. To evaluate the training and validation loss, we use the mean squared error.

3.1. Best frame selection

Estimating the 3D models requires the location of the container in video frames. We thus apply Mask R-CNN [18] trained on the MS COCO dataset [19] to each frame of the video and detect at most three objects among *bottle*, *wine glass*, *cup*, *book*, and *vase*. Among the detected objects, we discard ones with detection rates less than 50% and those with bounding boxes whose centers are not between one third and two thirds of the image width or between one third and two thirds of the height of the image height.

Sometimes the hand holding the container partially occludes it and makes mask segmentation difficult. In addition, the container is not always recognized and segmented; although we use the one with the highest detection rate to create the 3D model, it often detects multiple objects. Best frame selection is the algorithm to automatically select the frame that is the most visible for estimating the containers' 3D models.

At this stage, we compare two video frames $(V_f^{(k)}, V_f^{(l)})$ where f is a frame index and k and l are view indices that satisfy $(k, l) \in \{(1, 2), (1, 3)\}$. When multiple objects are detected and are close to each other, it can lead to a less accurate estimation of the 3D model of the container. To measure the loss caused by this, we introduce the penalty $p_f^{(v)}(i, j)$ at frame f of view v: defined as the number of pixels that lie within both the bounding box with the i-th largest detection rate and that with the j-th largest detection rate. As no more than three objects are detected, the total penalty $TP_f^{(v)}$, at frame f of view v, is obtained as:

$$TP_f^{(v)} = 2\left(p_f^{(v)}(1,2) + p_f^{(v)}(1,3) + p_f^{(v)}(2,3)\right). \tag{3}$$

To quantitatively evaluate the frames, we introduce View Score:

$$S_f^{(v)} = N_f^{(v)} - \sum_{v' \in \{k,l\}} TP_f^{(v')} \tag{4}$$

where $N_f^{(v)}$ is the number of pixel points in the container mask. Better visibility leads to larger $N_f^{(v)}$, and less ambiguous mask segmentation results in smaller $TP_f^{(v')}$. Hence, larger values of S_v are preferred. We finally select the best frame f^* by:

$$f^* = \underset{f \in \{1, \cdots, F\}}{\arg\max} \left(S_f^{(k)} + S_f^{(l)}\right). \tag{5}$$

3.2. Capacity and dimensions estimation

To estimate the 3D models of containers, we use and modify an existing algorithm, LoDE [20], which reconstructs the model of rotationally symmetric objects by sampling circumferences at different heights around the 3D centroid. It estimates the 3D centroid from the 2D centroids of the object mask of two different views, and iteratively fits the model by reducing each radius until it lies within both masks or reaches the pre-defined minimum value $r_{\min}$. We apply LoDE to $(V_{f^*}^{(k)}, V_{f^*}^{(l)})$. When the container mask is not extracted well, for example, its contour is unlikely to be detected, and the centroid cannot be obtained. In such a case, we change (k, l) to the other pair and executed the best frame selection again. In each iteration, LoDE decreases a radius if the corresponding circumference lies outside either mask. When one of the masks has a problem such as a hand occlusion, however, it can underestimate the radius. Therefore, we decrease radii only if their circumferences lay within neither mask.

Let Q_c, r_c and h_c denote respectively the 3D points in the c-th lowest circumference, their radii and height, where $c \in \{1, \cdots, L\}$ with L being the number of circumferences. Since adjacent radii can be different, we approximate the volume between Q_c and Q_{c+1} by a conical frustum. Therefore, the total volume is approximated by:

$$V = \sum_{c=1}^{L-1} \frac{\pi}{3}(r_{c+1}^2 + r_{c+1}r_c + r_c^2)(h_{c+1} - h_c). \tag{6}$$

To smooth the 3D model, we compute the difference $d_c = |r_c - r_{c+1}|$, and discard r_c if $d_c > d_{\max}$ for pre-defined $d_{\max}$. Let n', r'_i and d'_i denote, respectively, the number of the remaining radii, the i-th lowest one among them, and the difference $|r'_i - r'_{i+1}|$. This algorithm tends to estimate radii outside the mask to be much smaller than that inside the mask. Therefore, drastic changes in radii can be observed outside the mask; in other words, d'_i is greater when r'_i is outside the mask. Hence, we compute indices i_b and i_t from:

$$i_b = \min_{i \in \{2, \cdots, n'\}} \{d'_{i-1} > d'_i\} \tag{7}$$

$$i_t = \max_{i \in \{1, \cdots, n'-1\}} \{d'_i < d'_{i+1}\} \tag{8}$$

and consider $2r'_{i_b}$ and $2r'_{i_t}$ as the bottom width w_b and top width w_t; the height h is thus estimated by $h = h_{i_t} - h_{i_b}$.

3.3. Mass estimation

As discussed in Subsection 3.1, a hand occlusion can chip the mask. To reduce its impact, we restore partially occluded parts by utilizing the symmetrical property of the mask in the following way.

After the best frame selection, we extract the bounding box of the mask from $V_{f^*}^{(1)}$ with a padding of 15 pixels. In the extracted image, the lowest points in the mask correspond to the bottom of the container. Since the container has rotational symmetry, the mask is axially symmetric. We thus estimate the symmetrical axis is the perpendicular bisector of the segment connecting two edge points of the lowest points in the mask. Once the symmetrical axis has been determined, for pixels whose symmetrical counterparts are in the mask, we replace their values with those of the mask, which restores partially occluded parts.

We resize the restored mask to $(112, 112)$ to process in our CNN model. The model, depicted in Figure 2, is composed of four 3×3 convolutional layers with a padding of size 1, and three FC layers. Each convolutional layer is followed by the ReLU activation function, batch normalization, and 2×2 max-pooling layers, and each FC layer by the ReLU activation function and batch normalization layer. After the second FC layer outputs a tensor u, we concatenate u with (w_t, w_b, h).

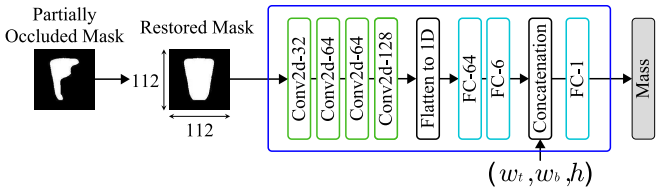

Fig. 2. Overview of the model for container mass estimation. ReLU activation functions, batch normalization, and max-pooling layers are abbreviated. KEY – Conv2d: 2D Convolution, FC: Fully Connected layer.

Table 1. Parameters settings. KEY – N_{mel}: number of mel filter banks, L: number of circumferences in the 3D model estimation, N: number of 3D points in a circumference, $r_{\max}$: maximum radius of circumferences, $r_{\min}$: minimum radius of circumferences, $d_{\max}$: maximum difference adjacent radii can take.

Parameter	N_{mel}	L	N	$r_{\max}$ (mm)	$r_{\min}$ (mm)	$d_{\max}$ (mm)
Value	128	145	19	75	2.5	1

4. EXPERIMENTS

4.1. Dataset

We used the CORSMAL Containers Manipulation dataset [21], which contains audio and RGB recordings of 15 containers but only data of nine containers are available in the training set; the rest is kept for the test set.

For filling level and type classification, we employed 7-fold cross-validation; we equally distributed the audio files of each configuration (i.e., level and type) to seven subsets and used one of them for the validation in each round. For the container capacity, dimensions, and mass estimation, we used 80% of the dataset and the rest for the validation of the mass estimation.

4.2. Implementation details

Table 1 summarizes the parameter values used in the experiments. We trained our model for 150 epochs in the filling level and type estimation, and 5000 epochs in the container mass estimation with the early stopping of patience 10; we stopped the training when the loss continued to increase for 10 epochs. In each epoch of the filling level and type estimation, the model first trained the two encoders with H_t and processed the type estimation while fixing the parameters of H_l. Then, it fixed the parameters of H_l and trained the two encoders with H_l for the level estimation.

4.3. Evaluation metrics

For each container, we evaluated the Mean Percentage of class-wise F1-Scores (MPF1) in the filling level and type estimation, and the Mean Absolute Percentage Error (MAPE) in the container capacity, dimensions, and mass estimation. The MAPE is the mean of the relative errors in percentage.

4.4. Filling type and level classification

Table 2 shows the estimation scores. The filling type and level were estimated with high MPF1, on average, more than 81% in both estimations. Figure 3 shows the confusion matrices of the estima-

Table 2. Scores for the estimations. An upward arrow next to each metric indicates the higher the better, while a downward arrow means the opposite. KEY – MPF1: Mean Percentage of class-wise F1-Scores, MAPE: Mean Absolute Percentage Error.

Container	Filling Level	Filling Type	Capacity	Top Width	Bottom Width	Height	Mass
Metric	MPF1 ↑	MPF1 ↑	MAPE ↓	MAPE ↓	MAPE ↓	MAPE ↓	MAPE ↓
Red cup	82.3	66.7	34.2	20.1	17.9	35.4	181.9
Small white cup	91.5	100.0	42.8	22.5	34.3	35.1	646.6
Small transparent cup	86.9	100.0	133.5	36.9	53.1	51.6	454.3
Green glass	81.4	100.0	61.8	26.8	35.8	38.7	29.6
Wine glass	78.0	100.0	49.0	21.9	54.1	20.0	21.2
Champagne flute glass	65.0	100.0	235.3	55.3	50.7	26.6	65.3
Cereal box	58.3	60.2	52.3	46.9	59.8	33.5	30.5
Biscuit box	86.7	84.1	54.5	25.1	46.5	22.5	65.4
Tea box	100.0	100.0	30.5	28.3	39.3	13.7	147.9
Average	81.1	90.1	77.1	31.53	43.5	30.8	182.8

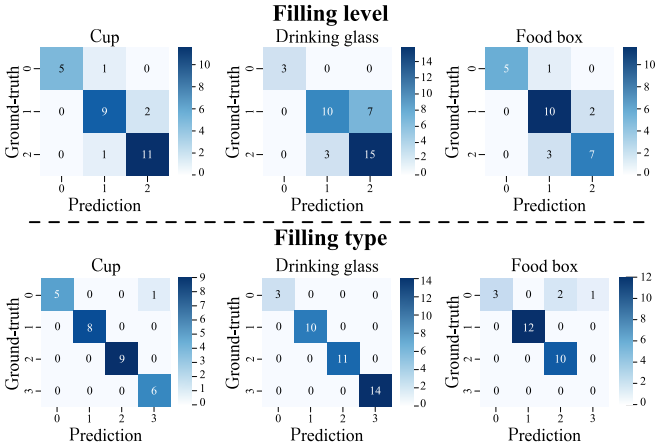

Fig. 3. Confusion matrices in the filling level and type estimation aggregated by the type of containers. They were computed from the validation data of the cross-validation in the round when the loss was the least. Level 0, 1, and 2 are *empty, half full* and *full*, while Type 0, 1, 2, and 3 are *no content, pasta, rice* and *water*, respectively.

tions, computed from the validation data of the cross-validation in the round when the loss was the least. Our model classified the filling types well; a few misclassifications were observed for containers with no filling. Comparing the estimated filling level and type by the container type, the model was likely to confuse *half full* and *full*, even though it correctly classified, or knew, the filling type. The CORSMAL challenge organizer computed other scores, shown in Table 3, on the public and private dataset using their own metrics [2]. We observed that the higher the filing level score, the higher the filling type score for all three datasets. This implies the shared architecture and parameters of the model learned fundamental but essential features in audio and transformed them into concrete classification results by each classification head.

4.5. Container capacity, dimensions, and mass estimation

Our algorithm estimated the capacity of champagne flute glasses least accurately. One of the reasons is that its colorlessness and transparency made the mask segmentation inaccurate. Besides, we fixed L for all the containers even though the height varies from container to container. Therefore, the taller the container is, the less accurate the 3D model estimation tends to be, especially when it is cone-shaped. This is because cone shapes require tracking the change in radii more precisely, which explains why the capacity of wine

Table 3. Estimation scores on the public and private test datasets computed by the organizer of the CORSMAL challenge using their own metrics. The higher the better. KEY – Pub: Public, Pvt: Private.

Task	Filling Level	Filling Type	Capacity	Top Width	Bottom Width	Height	Mass
Pub	68.14	83.56	71.95	73.23	61.81	78.58	43.61
Pvt	63.21	77.84	72.57	64.96	57.67	61.56	36.77

glasses, which are also colorless and transparent but shaped more like a cylinder, was better estimated.

The top and bottom widths, and heights were not influenced by this impact, because they depended only on the radii of the highest and lowest circumferences. Among them, the bottom width was estimated worst. Since View 2 and 3 look down at the manipulation, they are unlikely to see the bottom of the container, which made the 3D model estimation near the bottom inaccurate. Despite the different metric, the same tendency was observed in the scores computed for the public and private test datasets. While box-shaped containers do not have rotational symmetry, their capacities, widths, and heights were better estimated than those of other types, even though we estimated their shape by a conical frustum.

Inaccuracy of the mass estimation is closely related to that of the estimation of the dimensions. Moreover, the model's input is resized to a square, regardless of the shape of the restored mask image. Hence, information loss occurs when it is not square-shaped.

5. CONCLUSION

Filling level estimations of different filling types may be better handled if combined with type estimations. We have explored the joint filling level and type estimation in our Transformer-based model, which shares the same architecture and parameters in the two estimations except for the classification head. It achieved more than 81% in the mean of class-wise F1-Scores in both estimations. The accuracy of the algorithm and model proposed for capacity, dimensions, and mass estimation of containers depended largely on the accuracy in the 3D model estimation.

In future works, tuning the classification heads can be explored in filling level and type classification. As for container capacity, dimensions, and mass estimation, the 3D model would likely be estimated more accurately by tuning the number of circumferences according to the container height. Besides, changing the approximation formula of capacity according to the container type, and adjusting the padding size might also improve the accuracy.

[2] https://corsmal.eecs.qmul.ac.uk/challenge.html

6. REFERENCES

[1] Valerio Ortenzi, Akansel Cosgun, Tommaso Pardi, Wesley P Chan, Elizabeth Croft, and Dana Kulić, "Object handovers: a review for robotics," *IEEE Transactions on Robotics*, 2021.

[2] Ricardo Sanchez-Matilla, Konstantinos Chatzilygeroudis, Apostolos Modas, Nuno Ferreira Duarte, Alessio Xompero, Pascal Frossard, Aude Billard, and Andrea Cavallaro, "Benchmark for human-to-robot handovers of unseen containers with unknown filling," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1642–1649, 2020.

[3] Patrick Rosenberger, Akansel Cosgun, Rhys Newbury, Jun Kwan, Valerio Ortenzi, Peter Corke, and Manfred Grafinger, "Object-independent human-to-robot handovers using real time robotic vision," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 17–23, 2020.

[4] Min Wu, Bertram Taetz, Ernesto Dickel Saraiva, Gabriele Bleser, and Steven Liu, "On-line motion prediction and adaptive control in human-robot handover tasks," in *2019 IEEE International Conference on Advanced Robotics and its Social Impacts (ARSO)*. IEEE, 2019, pp. 1–6.

[5] Wei Yang, Chris Paxton, Arsalan Mousavian, Yu-Wei Chao, Maya Cakmak, and Dieter Fox, "Reactive human-to-robot handovers of arbitrary objects," *arXiv preprint arXiv:2011.08961*, 2020.

[6] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4561–4570.

[7] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox, "Deepim: Deep iterative matching for 6d pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 683–698.

[8] Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[9] Roozbeh Mottaghi, Connor Schenck, Dieter Fox, and Ali Farhadi, "See the glass half full: Reasoning about liquid containers, their volume and content," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1871–1880.

[10] Apostolos Modas, Alessio Xompero, Ricardo Sanchez-Matilla, Pascal Frossard, and Andrea Cavallaro, "Improving filling level classification with adversarial training," in *2021 IEEE International Conference on Image Processing (ICIP)*. 09 2021, pp. 829–833, IEEE.

[11] Chau Do and Wolfram Burgard, "Accurate pouring with an autonomous robot using an rgb-d camera," in *International Conference on Intelligent Autonomous Systems*. Springer, 2018, pp. 210–221.

[12] Hongzhuo Liang, Shuang Li, Xiaojian Ma, Norman Hendrich, Timo Gerkmann, Fuchun Sun, and Jianwei Zhang, "Making sense of audio vibration for liquid height estimation in robotic pouring," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5333–5339.

[13] Reina Ishikawa, Yuichi Nagao, Ryo Hachiuma, and Hideo Saito, "Audio-visual hybrid approach for filling mass estimation," in *International Conference on Pattern Recognition*. Springer, 2021, pp. 437–450.

[14] Vladimir Iashin, Francesca Palermo, Gökhan Solak, and Claudio Coppola, "Top-1 corsmal challenge 2020 submission: Filling mass estimation using multi-modal observations of human-robot handovers," in *International Conference on Pattern Recognition*. Springer, 2021, pp. 423–436.

[15] Guido Van Rossum and Fred L Drake Jr, *Python reference manual*, Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[16] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[20] Alessio Xompero, Ricardo Sanchez-Matilla, Apostolos Modas, Pascal Frossard, and Andrea Cavallaro, "Multi-view shape estimation of transparent containers," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 2363–2367.

[21] Alessio Xompero, Ricardo Sanchez-Matilla, Mazzon Riccardo, and Andrea Cavallaro, "Corsmal containers manipulation (1.0) [data set]," 2021.