2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW) | 979-8-3503-0261-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSPW59220.2023.10193633

SYMBIOTIC ARTIFICIAL INTELLIGENCE: ORDER PICKING AND AMBIENT SENSING

Zhe Ming Chng, Calix Tang, Darshan Krishnaswamy
Haoyang Yang, Shivang Chopra, Jon Womack, Thad Starner

School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA

ABSTRACT

Using egocentric video and head motion data from 67 order picking tasks (244 picks;149 orders), we learn visual models of the 10 objects picked to fulfill the orders. Boundary segmentations of the four actions (pick, carry, place, carry empty) of order picking had an average test RMSE of 1.11 seconds using computer vision and 5.53 seconds using only head motion (≈39.8 seconds/task). The 10 objects were clustered with 93.8% accuracy using weak supervision provided by the picks (which could occur in any order) specified in the tasks. We apply the 10 resulting models on independent test data to recognize three objects involving 50 tasks (185 picks;98 orders) and 10 objects involving 10 tasks (35 picks;24 orders). Accuracy was up to 90.3% and 69.1%, respectively. We propose order picking as a practical use case of egocentric Symbiotic AI, where ambient sensing is used without explicit supervision to train an agent which can then help the user improve task speed and accuracy.[1]

Index Terms— ambient AI, symbiotic AI, wearable sensing, egocentric vision

1. ORDER PICKING AIDED BY SYMBIOTIC AI

Symbiotic Artificial Intelligence (Symbiotic AI) is the idea that humans and AI may "live" alongside one another for mutual benefit. For example, the AI may watch a human partner's actions over time, learn how to do an activity correctly, and then help speed the activity and monitor for errors. To prove this concept in a practical scenario that has commercial relevance, we focus on the order picking problem.

Order picking is the process of selecting inventory items from pick bins and sorting them into place (order) bins for distribution. Roughly US$1 trillion in goods are distributed from almost a million warehouse sites each year. As much as 55% of all operating expenses for a warehouse are from the cost of order picking [1]. Additionally, 80% of all order picking in warehouses is done manually as robots do not yet have the dexterity to handle the variety of parts on most pick lines. As errors can lead to stopped assembly lines or customer dissatisfaction, manual pickers verify their picks by scanning barcodes either on the object itself or, if the object is too small, on the bin containing the object. However, the barcode scanning process can almost double the amount of time to complete a pick, cause physiological strain on the picker and does not eliminate all errors [2]. According to the CEO of Ox, a logistics company specializing in order picking using wearable computers, replacing barcode scanning with computer vision to identify the object as it is being picked would significantly improve both speed and accuracy [3].

Many industrial wearable computers, such as Google Glass EE2 and Vuzix M300, have cameras that could be used for this task. However, manually creating visual models of 100,000 objects in a warehouse would be expensive and onerous. Instead, can a wearable computer observe a picker's actions while fulfilling orders and automatically create the needed models? For example, the picker completes a task, selecting objects from a rack of pick bins for several orders and then placing those objects in place bins with one place bin for each order. If we can use action segmentation to segment carrying the object from picking and placing, we can then segment the hand with the object(s) and create a visual model of the object. Even though there may be several objects involved in the task and the picker may do the picks out of order, we can try all combinations of possible mappings of visual models to the picks in the task and select the mapping that provides the greatest consistency (and least variance in the visual model). Since pickers are often over 99% correct in their picks, incorrect picks will have little effect on the visual model over time. In this manner, a visual model can be learned with no explicit training, and, as the model becomes reliable, the system can begin to alert the picker when a pick may be incorrect. Similarly, if a customer complains about an incorrect order, the same visual model might be used to resolve disputes by finding images of the objects being placed into the customer's shipment.

Placing sensors throughout a warehouse is impractical from an expense, deployment and maintenance perspective. Instead, we focus on adding ambient multimodal sensors to the picker's clothing. Specifically, we use a wide field of view camera with an inertial measurement unit mounted on the picker's forehead facing toward the hands to give a egocentric view. In industry, a head worn wearable computer with display and camera (such as Glass EE2) might be retrofitted with a small mirror or fish-eye lens so as to be able

[1]Code & data: `github.com/czming/ai-through-symbiosis`

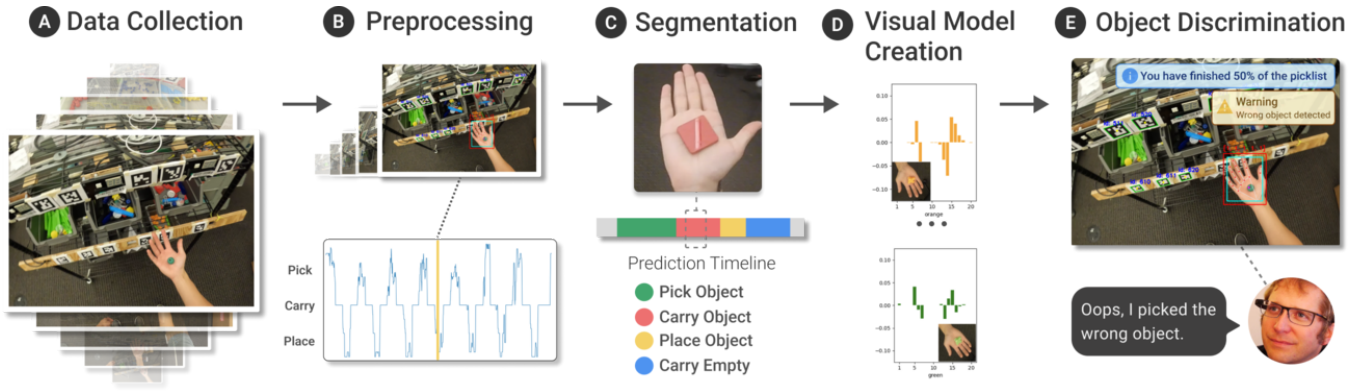

Fig. 1: Pipeline for generating visual models of objects from observing picker behavior.

to track the hands.

In the 1990s, Starner [4] hypothesized that wearable computers that "see" and "hear" what the user does would be able to leverage this egocentric perspective to sense general context from the first-person perspective of the user and, in turn, learn to help the user in everyday scenarios. Initial work on such symbiotic AI showed progress in discovering semantically meaningful locations and predicting movement between them [5], face recognition [6] and social engagement [7]. Roboticists have also been interested in wearable computing and first person perception [8] with early work focusing on learning human actions and mapping them on to humanoid robots [9]. However, the last decade has shown accelerating progress as wearable sensors and machine learning methods have advanced [10, 11, 12], especially for set-supervised action segmentation [13, 14, 15]. Most recently, with the release of the Ego4D dataset in 2022 [16], there has been a surge of interest in data gathered from a first-person viewpoint. Ego4D identifies several types of questions that might be posed with such data related to episodic memory, hand and object state, audio/video diarization, social interactions, and forecasting. Few of these dataset are both motivated by immediate, practical commercial use cases and have enough repetition such that unsupervised or weakly supervised learning can be used to recover visual object models.

2. VISUAL MODEL LEARNING PIPELINE

Fig. 1 shows our pipeline for deriving a visual model for each picked object and then using that model to recognize objects in unseen picking activities. The pipeline produces *action segmentations* and pseudo-label assignments from *action sets* and ambient sensor data.

Our order picking "warehouse" was based on those found in automobile manufacturing lines but simplified to allow for rapid data generation. The picking shelving unit has 12 pick bins, each with a unique object. A shelving unit with place bins for three orders is about five steps from the pick bins. Both shelves had AruCo markers [17] associated with the bins, similar to the barcodes found in industrial scenarios.

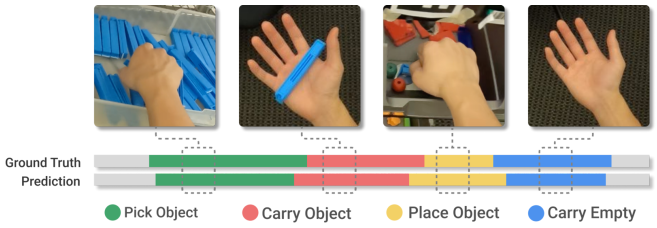

Fig. 2: Predicted action segmentations

We generate a set of picklist *tasks* such that each picklist contains an average of 3.62 objects to be picked with at most 12 possible orders in which the objects could be picked (ignoring repeated object labels). Tasks are randomly sampled for each of three pickers (37, 20, and 20 tasks) to complete. Each picker is fitted with a GoPro Hero 10 capturing 30FPS video at 4k and accelerometer and gyroscope data at 200Hz.

We detect ArUco markers [17] in each image frame using the OpenCV library [18]. To differentiate between picks and places, we give ArUco markers associated with pick bins a score of +1 and markers associated with place bins a score of -1. The sum is a feature for the next stage of processing by hidden Markov model (HMM). An example can be seen in Figure 1B where a positive number indicates a likely pick, zero indicates a probable carry, and negative numbers are associated with places. Head tilt and yaw is extracted from the gyro data and downsampled to match the rate produced from the lower frequency ArUco data. We use the Mediapipe Hands library [19] to extract hand keypoints from the image. We crop a rectangular portion of the image which contains the entire hand as seen by the blue bounding box in Fig. 1B. We use a Gaussian convolution step to smooth our features across adjacent frames. After preprocessing and smoothing the data, we employ HTK, a HMM toolkit [20] and tooling from GT2K [21] to create action segmentations.

HTK allows for the specification of a rule-based grammar, which is well-suited to the highly structured task of order picking (i.e., "pick," "carry," "place," and "carry empty"). Picklist tasks are specified as objects and place bins, which implies a set of sequential actions if one ignores the identity

of the objects themselves. For example, a picklist with two objects would have the predetermined sequence of actions: pick, carry, place, carry empty, pick, carry, place, carry empty. These predetermined sequences enable a more precise alignment of observations to the known sequence of hidden states through forced-alignment. In this manner, a HMM for each of the four general actions is created (ignoring the identity of the object involved). When testing on an unseen sequence, Viterbi decoding is used to find the most likely segmentation of actions, outputting timestamps for action boundaries. Fig. 2 shows example output of the decoding process. Using cross validation with an 90% to 10% train-to-test split to ensure a "fair" test, we create segmentation boundaries for each of the picklist tasks and use these cross validation splits for training and testing the color histogram visual object models in the next section. The HMM topology is a six-state left-to-right topology with no skip states. Transitions are initialized with equal probability for self-loops and moving forward. Emission probabilities are single gaussians per dimension initialized with mean 0 and standard deviation of 1.

We use the boundaries obtained above to identify sequences of frames where the picker is carrying the object in the hand ("carry" frames). To find a representation of each sequence of carry frames, we take the average histogram of hue values (each bin representing 2 degrees of hue, totalling 180 bins) of pixels within the segmented hand portion of the image across the frames in the sequence. To remove variability due to skin color of the pickers' hands, we remove the average histogram of hue values across the sequence of "carry empty" hand images. We use this net histogram of hue values as a vector to represent each carry frame sequence.

Next, we must obtain labels for the object being carried in each carry frame sequence. We first randomly assign object labels to carry frame sequences with the constraint that the picks must be consistent with the picklist task (i.e. counts of each object must match those in the picklist). We improve on the clustering iteratively using a simulated annealing method by looking at the distance of vectors from their cluster mean, normalized by the standard deviation of points in the cluster along each respective dimension. To reduce the time complexity of each comparison, we reduce the number of dimensions of hue that we compare from the original 180 bins to 12 bins. In each iteration we select a random pick from each picklist. Then, we iterate through all the other picks and see if there is another object where swapping the labels on the two objects reduces the normalized distance of the vectors from their cluster means, where the cluster means are computed without the inclusion of the two picks under evaluation. If the normalized distance is not reduced by any potential swaps with objects in the picklist, we use simulated annealing to assign a probability p of swapping to avoid local minima. We reduce p proportionately with the number of epochs. When deciding with which specific pick to swap, we assign the probability based on the distance reduction achieved by the different potential swaps. We repeat this method across all the different picklists and for $n = 500$ epochs.

The resulting object model consists of the average histogram of hue values across the various sequence of frames where the picker is predicted to be carrying the same type of object. At inference time, we predict the object type whose model has the closest hue histogram to the test object.

3. EXPERIMENTAL RESULTS AND DISCUSSION

Experimental setup: For the 77 10-object picklist tasks collected in Section 2 we split our dataset into a train/validation /test split of 60/7/10. Hyperparameter tuning is performed using three fold cross-validation of the models being trained on 60 tasks and tested on 7 validation tasks. The 67 tasks consisted of 244 picks and 149 orders. The final 10 tasks (35 picks;24 orders) in the independent test set are used to evaluate the performance of the representations learned during training. A seperate set of 50 tasks (185 picks;98 orders) using 3-object picklists provide another test of the system.
Results: In Fig. 3, we present the results on our training set on our evaluation metrics of accuracy for label assignment, represented using a confusion matrix. We define accuracy as the average of $\frac{N_{TP}}{N_P}$ across all object classes, where N_{TP} is the number of accurately assigned labels and N_P is the number of true labels. We include root mean square error (RMSE) as our metric for boundary detection accuracy, which is the RMSE between the predicted boundary timestamp and the actual boundary timestamp between different actions, and is represented by a histogram of RMSE achieved on each picklist when it was in the test set as part of k-fold validation, where $k = 40$ and with a $90\% - 10\%$ train-test split. We include a histogram of the RMSE values obtained on different different picklists when using i) ArUco score and ii) head motion data as the output from the HMM for training in Fig. 3. On the training set, our model achieves a clustering accuracy of 93.8% from the unordered set action labels. In Fig. 4, we include prediction results of the object models learnt on our two test sets: i) ten 10-object picklists ii) 50 three-object picklists, and a visualization of the average hue histogram of each object type in Fig. 5. On the ten picklists with 10 objects, we have an average RMSE of 1.11 seconds for our boundary predictions, with an accuracy of 69.1% when we constrain predictions to the set of classes that appear in the picklist and 56.6% without constraints. On the testing set with three objects, we achieve an accuracy of 90.3% with an RMSE of 1.53 seconds after adjusting for three outliers that had errors greater than three standard deviations away from the mean.

4. DISCUSSION AND FUTURE WORK

The 93.8% 10-object clustering accuracy and the 3-object test set accuracy demonstrate that the automatic creation of visual models by observing actions made during order picking

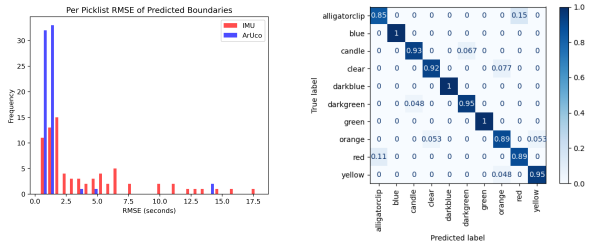

Fig. 3: Left: Histogram of boundary errors (67 picklists total) where red represents segmentations using head motion and blue represents ArUco markers. Right: Average confusion matrix of clustering across three folds of the training set.

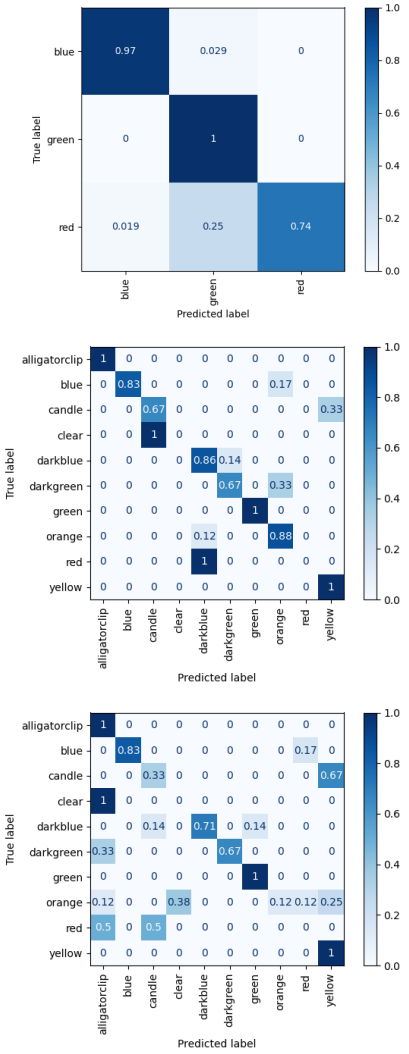

Fig. 4: Top: Object predictions on 3-object task. Middle: 10-object predictions constrained to the types of objects in the task. Bottom: 10-object predictions with no constraints.

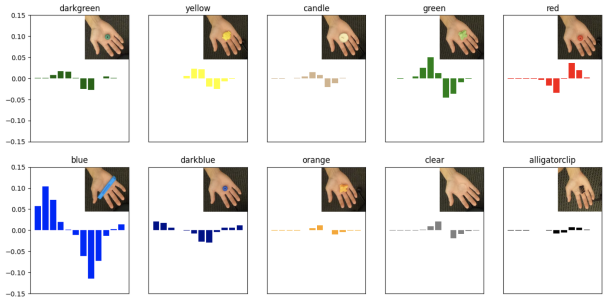

Fig. 5: Histograms of hues and saturations for the ten object classes based on predicted boundaries using ArUco markers.

is possible. While 90.3% and 1.53 sec accuracy in finding and labelling picks may be sufficient to aid in providing proof to a disgruntled customer that the correct objects were delivered, it is not sufficient for alerting a picker that a task was picked incorrectly in real-time. Perhaps more data is needed to provide better models? The difference in accuracy the 10-object tasks (constrained by the set of unique classes in the task picklist versus not constrained) suggests under training or a model that is not sufficiently expressive for the task.

To this end, we are gathering more data and investigating two different approaches: an end-to-end deep learning approach using a triplet loss and Siamese network method inspired by FaceNet [22] to learn embeddings for objects before using our clustering step to map embeddings to object classes and an approach that uses the labels predicted for each carry frame sequence and trains a ResNet-18 [23] model using training examples based on those labels. Preliminary results for the former approach shows 81.3% clustering accuracy while the latter approach attains a prediction accuracy of 75.7% on our 10-object test set.

Another limitation of the current approach is the dependence on hand detection to segment the area of interest, which limits the applicability of the pipeline to objects with substantial hand occlusion. We are exploring depth-map and optical flow-based approaches for scale, viewpoint, and illumination invariance in our approach.

5. CONCLUSION

Our study demonstrates the potential of egocentric Symbiotic AI in the context of order picking tasks. With minimal supervision, our approach accurately segmented the four actions of order picking and learned visual models of the 10 objects, achieving promising accuracy in recognizing the objects in independent test data. These findings suggest that egocentric video and head motion data can be leveraged to develop practical applications for ambient sensing and may enhance the speed and accuracy of order picking in the future.

6. ACKNOWLEDGEMENTS

We would like to thank Cisco for funding this research.

7. REFERENCES

[1] Kees Jan Roodbergen René de Koster, Tho Le-Duc, "Design and control of warehouse order picking: A literature review," *European Journal of Operational Research*, pp. 481–501, October 2007.

[2] Charu Thomas, Theodore Panagiotopoulos, Pramod Kotipalli, Malcolm Haynes, and Thad Starner, "RF-pick: comparing order picking using a HUD with wearable RFID verification to traditional pick methods," in *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, Singapore Singapore, Oct. 2018, pp. 168–175, ACM.

[3] Charu Thomas, "Personal correspondence," 2020, Internal Discussion, Ox CEO.

[4] Thad Starner, *Wearable computing and contextual awareness*, Ph.D. thesis, Massachusetts Institute of Technology, 1999.

[5] Daniel Ashbrook and Thad Starner, "Using gps to learn significant locations and predict movement across multiple users," *Personal and Ubiquitous computing*, vol. 7, pp. 275–286, 2003.

[6] Bradley Singletary and Thad E Starner, "Symbiotic interfaces for wearable face recognition.," in *Proc. of Human Computer Interaction International*, New Orleans, LA, 2001, HCII, pp. 813–817.

[7] Bradley Singletary and Thad E Starner, "Learning visual models of social engagement," in *Proc. ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*. IEEE, 2001, pp. 141–148.

[8] Rodney A Brooks, "Intelligence without reason," in *The artificial life route to artificial intelligence*, pp. 25–81. Routledge, 2018.

[9] Charles Kemp et al., *The acquisition of inductive constraints*, Ph.D. thesis, Massachusetts Institute of Technology, 2008.

[10] Hilde Kuehne, Ali Arslan, and Thomas Serre, "The language of actions: Recovering the syntax and semantics of goal-directed human activities," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 780–787.

[11] Alireza Fathi, Ali Farhadi, and James M. Rehg, "Understanding egocentric activities," in *2011 International Conference on Computer Vision*, 2011, pp. 407–414.

[12] David Minnen, Irfan Essa, and Thad Starner, "Expectation grammars: Leveraging high-level expectations for activity recognition," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* IEEE, 2003, vol. 2, pp. II–II.

[13] Mohsen Fayyaz and Jurgen Gall, "Sct: Set constrained temporal transformer for set supervised action segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 501–510.

[14] Jun Li and Sinisa Todorovic, "Set-constrained viterbi for set-supervised action segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10820–10829.

[15] Alexander Richard, Hilde Kuehne, and Juergen Gall, "Action sets: Weakly supervised action segmentation without ordering constraints," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 5987–5996.

[16] Kristen Grauman et. al, "Ego4d: Around the world in 3,000 hours of egocentric video," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 18973–18990.

[17] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[18] G. Bradski, "The OpenCV Library," `https://opencv.org`, 2000–2021, Accessed: 4 Sep 2022.

[19] Camillo Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," *arXiv:1906.08172 [cs]*, June 2019, arXiv: 1906.08172.

[20] Steve J Young and SJ Young, "The htk hidden markov model toolkit: Design and philosophy," 1993.

[21] Tracy Westeyn, Helene Brashear, Amin Atrash, and Thad Starner, "Georgia tech gesture toolkit: supporting experiments in gesture recognition," in *Proceedings of the 5th international conference on Multimodal interfaces*, 2003, pp. 85–92.

[22] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.