Egocentric Prediction of Hand-Object Interaction

Luis Guzman, Isaac Kasahara, Aditya Rajguru, and Helena Shield

Abstract— **Predicting probabilities of hand-object interaction from an egocentric point of view will prove to be useful in the developing fields of augmented reality and human-robot interaction. In this paper, we propose a method to predict hand-object interaction from RGB images captured from a first person perspective. Our proposed method aims to improve upon previous methods by allowing for egocentric camera movement and by using alternative methods to detect objects and predict trajectories. Our contact prediction pipeline starts by locating the objects in the image via the Detectron2 implementation of Mask RCNN. Hand locations and automatic contact labels were generated using a network pretrained on detecting hand-object interaction. These locations are then preprocessed to account for camera movement before being passed into an LSTM to calculate the predicted trajectory of the hand. The trajectory and detected object locations are then used as inputs to a MLP in order to predict the probabilities that the hand interacts with these objects. Our data show that we successfully reproduced the results of previous work on stationary video and our method is able to be applied to egocentric video without a significant accuracy drop. We evaluated our method by counting video frames that result in a correct prediction and found our method to be 80-88% accurate depending on the number of objects in the scene and the magnitude of camera movement.**

I. Introduction

In the growing fields of human-robot interaction and virtual reality, the ability to predict human behavior is becoming increasingly important. Workplace safety demands that any person working alongside a robot be not at any additional risk for injury [15]. As hospitals, warehouses, and factories continue to integrate robots into their workforce, new technology is developing to keep human workers safe. One of these approaches, prediction of human behavior, can allow robots to avoid contact or give warnings in risky situations. Virtual reality can also use similar technology to improve interactive environments [7][16]. However, much of the work in human motion prediction has only been done from a stationary third party view, and cannot be easily translated to an egocentric one. Since many applications that demand humans closely interface with robots will require the use of an egocentric camera, it is important that this adjustment be made. Previous work in this area has dealt with hand motion prediction from a third party view [10]. Tao et al. developed a model that would take in an RGB video of a hand reaching towards multiple objects and predict which one the hand would come in contact with. In this paper, we propose a model that can accomplish the same task from an egocentric point of view.

The egocentric viewpoint creates a distortion in the appearance of motion of the scene. Head movements make stationary objects appear as though they are moving [3]. However, not every subject in view is affected in the same way. Since hands are attached to the egocentric body, their motion is not independent of the egocentric camera motion. In addition to this, objects and hands can move in and out of frame as the head turns in different directions. Finally, the egocentric view may raise issues with occlusion as hands overlap objects without contact more often than in stationary video.

Our method exploits the temporal nature of the problem in order to predict future trajectories of the hand. We make use of existing work in order to preprocess datasets and label them automatically and also account for the egocentric camera motion. We then train a Recurrent Neural Network (LSTM) to predict the trajectory of the hand to the objects and also use a fully connected neural network to predict the probability of the hand making contact with the object based on the predicted trajectories.

II. Baseline Method

Previous work by Tao et al. has shown that predicting hand-object contact is possible when using a stationary camera [10]. The network proposed in their paper uses YOLOv3 for object detection followed by a Social LSTM for trajectory

prediction. Lastly, they use a convolutional neural network (CNN) to predict the probability of hand-object contact based on the object locations and hand trajectories. Their model is able to accurately predict hand-object contact, even when the human intends to pass over one object to contact another. Their method did not include quantitative results, so for our comparison, we plan to compare the accuracy of a stationary camera to an egocentric one. If we can achieve nearly the same results as a stationary camera, we will consider our model to be a success. Due to the efficacy of their method, we choose to break down the egocentric problem into similar steps: object detection, trajectory prediction, and probability of contact. A limitation of their method is that it cannot generalize to a moving camera. To account for this, an additional preprocessing step is necessary. In the following paragraphs, we examine the state-of-the-art in each of these categories.

III. Related Work

Detectron2 is a system built by Facebook AI Research (FAIR) that implements many state-of-the-art object detection algorithms like Mask R-CNN and Cascade R-CNN [2][6]. Their baseline data shows a 2x performance increase over previous methods like Faster R-CNN and shows similar runtime performance to RetinaNet, while being slightly faster to train [11]. Another popular option, YOLOv3 exhibits highly efficient object detection but struggles in classification when compared to its competitors [22]. Image classification problems commonly use a convolutional neural network (CNN) for probability prediction [24][30]. Other approaches include compounding multiple classifiers and using them as a basis for an overall classification [25]. There are more recent algorithms such as M2Det and DetectoRS that could potentially offer better performance [8][12], but we chose to use Detectron2 Mask R-CNN due to its good overall performance, clear documentation and relative simplicity of the network architecture. Detectron2 is able to run in near-real-time at around 5-10 frames per second. We expect this stage to have the largest amount of computational overhead, so running object detection in real-time is essential to the computational efficiency of our final model. From the studies mentioned above, we expect this network to outperform the YOLOv3 detector that was used in Tao et al.'s baseline results.

For the task of trajectory prediction, Alahi et al. suggest using Social LSTMs to learn human behavior of walking in crowded spaces [1][20]. Another method uses Generative Adversarial Networks to achieve the same task [5]. Many of the models focused on hand gesture recognition also use dynamic probability LSTMs on gesture segmentations for classification[23]. A different paper shows promise using an LSTM for learning the movement of human arms for a specific task [13]. In this case, the LSTM learns the predicted trajectory by feeding previous joint information of the arm into the LSTM. For our purposes, we adapted this model to only take in the hand position, as we will not always have the full arm in frame due to the positioning of the camera. Sequential data classification applications also use ConvLSTMs [26]. For our approach, we used an MLP to predict probabilities of contact from the predicted trajectories.

Lastly, we needed a method to automatically label hand-object contact in our dataset for training the MLP and LSTM. Hand-object interaction is a challenging problem which has been studied in the context of action recognition and 3-D pose estimation. Most of the earlier research has relied on depth sensors to detect hand-object interaction, but our dataset and use cases do not include depth information, since we are using a RGB monocular camera. Recent research has aimed to create single shot end to end networks to detect 3-D pose of the hand and the objects as well as recognize the activities being performed by exploiting the temporal nature of RGB videos [17]. Other works have used optical flow features of body movements as a whole to predict actions [18]. Facebook's FAIR group has developed InteractNet, a method that uses human pose estimation to extract (human, verb, object) tuples when a person interacts with an object in video [4]. Their method uses Faster-RCNN for object detection, then they calculate the mean interaction location given the human's appearance and action. By minimizing the L1 loss between the predicted object location

and the detected objects in the scene, they can accurately predict human-object interaction. 100 Days of Hands (DOH) builds on this methodology by extending it to hand-object interaction [9]. We use the Days of Hands network to robustly detect hand-object interactions in a given frame and give us the position of the hand and state of contact with the object, which helps us to preprocess and label image frame data for training the LSTM and MLP [9].

IV. Data Collection & Labeling

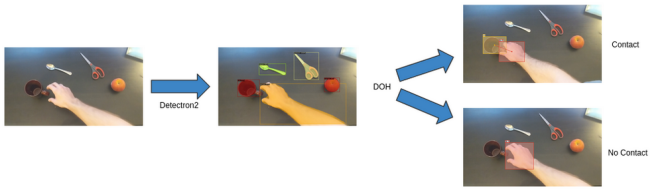

Fig. 1. Breakdown of object detection and labelling

We used a head mounted camera to collect video of our own hands interacting with objects recognized by the COCO dataset with varying levels of camera movement and occlusion, as well as varying numbers of objects. Originally we had planned on supplementing this data with videos from the Epic-Kitchens-100 dataset which also features an egocentric camera POV. Unfortunately, we ran into issues with a large portion of the clips containing objects that were not included in the training data, so we trained our model mostly on our own data and used some of the viable clips from Epic-Kitchens-100 for testing. Because neither our own dataset nor the Epic-Kitchens-100 videos came with contact labels and manual labeling would be impossible for a dataset this large, we developed scripts to automatically label video frames. First, we utilize the Mask R-CNN network in order to detect objects of interest in the scene. Then we use the DOH network proposed by Shan et. al. to localize the hand through its cartesian coordinates and determine at each frame whether contact was made.

V. Model

Our model uses Mask R-CNN in Detectron2 for object recognition, followed by a preprocessing stage to account for camera movement. Next, we implement a LSTM for trajectory prediction and finally use a MLP to predict the probability of contact. In the following paragraphs, we take a more in-depth look at each component of our contact prediction pipeline.

Object Detection: To get the positions of objects in the scene, we used a network in Detectron2 that is pretrained on the COCO dataset and is able to detect 80 common object categories. After running a video frame through the network, Detectron2 provides us with a list of bounding boxes and object labels for every object detected in the scene. We then calculate the centroid of each bounding box to use as our object position. One issue we ran into with our object detection was that Detectron2 would draw the bounding box around a person's entire arm or body instead of just their hand. This would result in inaccurate hand tracking since as more or less of a person's arm is visible in the scene, the bounding box centroid would change accordingly. We opted to instead use the hand bounding boxes given by the Days of Hands network, since this was specifically trained on recognizing hands. Once we made this switch, we got much more accurate hand tracking results that we could then preprocess to account for camera movement.

Because we're interested in how object positions change with time, we needed to develop a method of tracking which bounding boxes correspond to the same object from frame-to-frame. This required modifying the source code of Detectron2 and DOH to assign a unique object ID to each object. We used intersection over union (IoU) and the object labels to determine which instances belong to the same object. Essentially, if two instances have the same label and their bounding boxes overlap significantly in consecutive frames, we assign them the same object ID. By filtering the object positions for a single object ID, we can build a path trajectory for that object.

Position Preprocessing: Motion normalization is our largest divergence from the Tao et al. baseline, and it is a necessary step to account for a non-stationary camera. Because the camera will be moving, we can no longer rely on stationary objects to have the same pixel coordinates from frame to frame. In addition, the hands being an

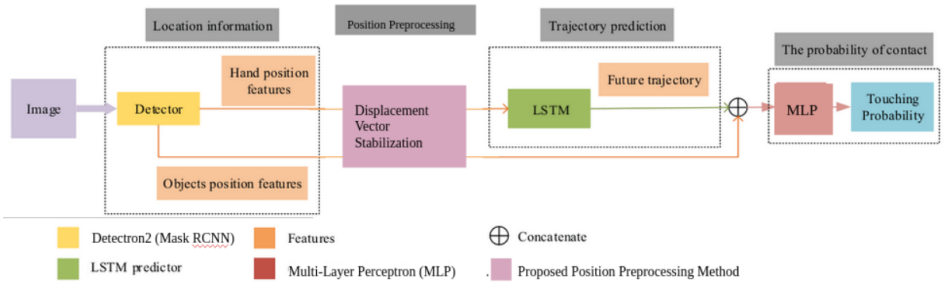

Fig. 2. Architecture of our proposed model (image adapted from Tao et al.)

extension of the egocentric body creates a unique situation where the camera movement effects on the object position differ from the effects on the hand position. Our method aims to reduce computational complexity by defining a vector between the centroids of the hands and each object in the scene, similar to [9]. Then this hand-object vector is given as the input into the LSTM instead of the raw position values. This method utilizes the assumption that contact can be defined as the distance vector trending towards zero, so we will train our MLP to recognize this behavior.

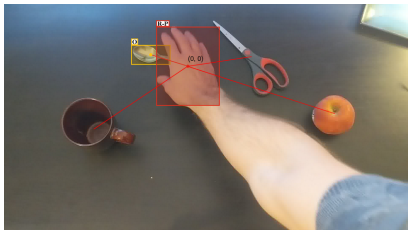

Fig. 3. An image showing four computed hand-object vectors to account for camera movement

Trajectory Prediction: Our second major divergence from the Tao et al. baseline is how we're implementing our trajectory prediction. Tao et. al opted to use a social LSTM which uses a social pooling layer to connect the LSTMs for each object in the scene [1]. This allows for the motion of one object to be taken into account when predicting motion of another. However, with our motion normalization approach, we decided on using a non social LSTM. Since the presence of one object does not significantly affect the movement of the hand to another, we do not believe the social pooling layer to be a beneficial addition to the standard trajectory prediction. We use an LSTM on every hand-object pair to predict future motion in the vector format described above. Our LSTM was trained using the past 8 relative hand positions to predict one time-step ahead, and was used recursively to predict multiple steps in the future. Due to the recursive nature of the prediction with the LSTM treating predictions like observations, the LSTM struggled when predicting more than 8 frames out into the future.

Another challenge we experienced was deciding how LSTM will handle objects that leave the frame. The LSTM needs an entire valid series to predict on, so if just one of those values is missing due to occlusion or detection failure, we cannot predict the trajectory. Due to time constraints, we have not accounted for missing values in the series-rather, series with missing values are not predicted on. This could later be improved on by interpolating missing values, but that method will still not be valid for objects that leave the scene. This issue could further be minimized by using a fish-eye lens with a wide field of view. Contact Probability Estimation: The predicted trajectories and object locations are then passed into the MLP to determine probability of hand/object contact. Based on our past observation of the trajectory, we take the predictions of the next 5 frames and pass the euclidean norm of the distance to the object as the input to our MLP. Our MLP consists of 3 layers with 10, 30 and 2 nodes respectively. All layers except the output layer have ReLu for activation, and the output layer has a softmax function for activation. The 2 nodes in the output layer correspond to the probability of no contact vs contact.

A challenge for this network is to differentiate object contact versus occlusion. The network needs to identify whether the hand plans to contact an

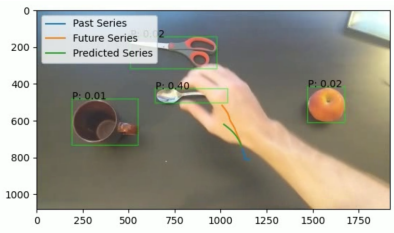

Fig. 4. An image showing an occlusion scenario. Here, the hand is passing over the spoon to contact the scissors. Our method accurately predicts no contact (40%) for the spoon but is not able to predict contact (2%) of the scissors.

object or just pass over one to contact another. Tao et al. uses the convolutional network to account for this, so we anticipated similar error rates with our method. However, there are inherently more occlusions in egocentric video due to hands taking up a larger portion of the frame. From our testing, the network is able to predict cases where occlusion does not result in contact, but more training examples are required to make this more robust.

VI. Results

Qualitative results (figure 5) show that our network can successfully predict hand-object contact based on hand trajectories. Our method for quantifying our method's performance was to count the number of frames where the network made a correct prediction of contact. We ran this test for 50 testing videos, which were held-out from our training set. Our testing dataset was also split into four categories corresponding to stationary and egocentric (moving) cameras and single and multiple objects in the scene. We found that our method was able to successfully reproduce the performance of Tao et al. on stationary video. Furthermore, our method was able to generalize to the case of egocentric video without a significant performance drop. The average performance for stationary video was 86.4% and the performance for egocentric video was 81.0%. We attribute this small performance decrease partially to the object detection algorithm, which cannot as accurately detect objects with motion blur, and partially to the LSTM, which would not give as accurate of a prediction if the input series is not as stable.

	Single Object Accuracy	**Multi Object Accuracy**
Stationary Video (baseline)	0.8447	0.8842
Egocentric Video	0.8202	0.7981

TABLE I

Quantitative results generated from counting video frames with a correct prediction

VII. Conclusion

Altogether, we created a system that successfully predicts hand-object contact in video taken from an egocentric point of view. We created a pipeline consisting of MaskRCNN for object detection, a preprocessing step to account for camera movement, and an LSTM to predict the future trajectory. This path is used in an MLP to get a probability for hand-object contact. Our results on stationary video show similar performance to the Tao et. al baseline, and we achieved similar overall accuracy when applying our method to egocentric videos. Future work would include adding padding and interpolation to the detected time series to make the system more robust against missing data points. Additionally, we believe that gathering additional training data would result in the LSTM making more accurate predictions further into the future.

VIII. Role

All members equally contributed to the data collection, processing, programming, and writing that this paper consisted of. Further details of each member's contributions are broken down below:

Luis Guzman: Collected 25 videos of hand-object interaction, created the automatic labeling (Detectron2/DOH/ObjectID) and final visualization scripts along with Aditya, wrote various sections of the paper and presentation

Isaac Kasahara: Collected 55 videos of hand-object interaction, wrote program to generate hand prediction LSTM, wrote various sections of the paper and presentation

Aditya Rajguru: Researched on hand object-detection networks, contributed to object detection and automatic labelling ideas and implementation, contributed to visualization scripts along with Luis and wrote some sections of the paper and presentation.

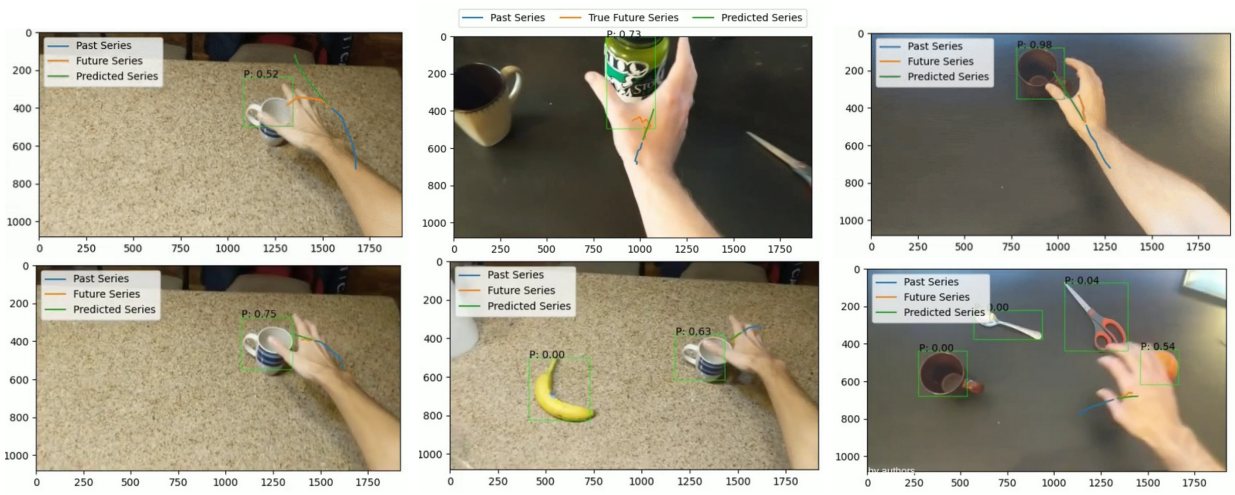

Fig. 5. Qualitative results of our method. The first image (top left) shows an extreme curved path. The LSTM is unsure where the hand is heading but still gets a correct prediction at 52%. The second image (top middle) shows a scenario with large camera movement. The next two show single object contact and the last two (bottom right) shows multi-object contact.

Helena Shield: Wrote MLP program to generate hand-contact predictions, researched and brainstormed motion-normalization methods, wrote various sections of the paper and presentation

IX. Comments from the Committee

Proposal Presentation:
How to deal with occlusion (see Model section)
Why image stabilization (see Model section)
Final Presentation:
Calculating Multiple Trajectories:
An alternative approach to this problem with our set up would be calculating many trajectories and using a monte-carlo method to determine the probabilities of contacting each object. Since we decided to focus on hand-object vector trajectories, implementing this would take some major changes to our LSTM and MLP but would make a good comparison.
Our 20 Frame Requirement:
Often when detecting over multiple frames, an object that appears throughout may not be detected in every single frame. While to a human eye the existence of this object is clearly continuous, the inherently discrete nature of data sampling from videos means that a missed frame interrupts a series of data points that makes it impossible for the LSTM to work. For this project we had sufficient amounts of data that we could simply enforce a 20 frame requirement but in the future works it would be beneficial to use interpolation to generate those lost points and increase the data that this system could train and operate on.

X. Links to Code and Dataset

Github: https://github.com/luigman/CSCI5561ProjectFall2020
Project Video: https://youtu.be/nKqXu4bZbFY
Dataset: https://drive.google.com/drive/folders/1zVOAfGqvG-TJOck1QsgvL5am6JCHrMsc

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei and S. Savarese, "Social LSTM: Human Trajectory Prediction in Crowded Spaces," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 961-971, doi: 10.1109/CVPR.2016.110.

[2] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into High Quality Object Detection," arXiv:1712.00726 [cs], Dec. 2017, Accessed: Oct. 28, 2020. [Online]. Available: http://arxiv.org/abs/1712.00726.

[3] O. Cohen, A. Apartsin, J. Alon and E. Katz, "Robust Motion Compensation for Forensic Analysis of Egocentric Video using Joint Stabilization and Tracking," 2018 IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE), Eilat, Israel, 2018, pp. 1-5, doi: 10.1109/ICSEE.2018.8646211.

[4] G. Gkioxari, R. Girshick, P. Dollár, and K. He, "Detecting and Recognizing Human-Object Interactions," arXiv:1704.07333 [cs], Mar. 2018, Accessed: Oct. 28, 2020. [Online]. Available: http://arxiv.org/abs/1704.07333.

[5] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks," arXiv:1803.10892 [cs], Mar. 2018, Accessed: Oct. 28, 2020. [Online]. Available: http://arxiv.org/abs/1803.10892.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," arXiv:1703.06870 [cs], Jan. 2018, Accessed: Oct. 28, 2020. [Online]. Available: http://arxiv.org/abs/1703.06870.

[7] Dengwu Ma, X. Lv, Wen Ye and Xiaoyan Qu, "Research of Human Head Motion Prediction and Tracking in Virtual Environment," 2006 6th World Congress on Intelligent Control and Automation, Dalian, 2006, pp. 1621-1625, doi: 10.1109/WCICA.2006.1712626.

[8] S. Qiao, L.-C. Chen, and A. Yuille, "DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution," arXiv:2006.02334 [cs], Jun. 2020, Accessed: Oct. 28, 2020. [Online]. Available: http://arxiv.org/abs/2006.02334.

[9] D. Shan, J. Geng, M. Shu, and D. F. Fouhey, "Understanding Human Hands in Contact at Internet Scale," arXiv:2006.06669 [cs], Jun. 2020, Accessed: Oct. 28, 2020. [Online]. Available: http://arxiv.org/abs/2006.06669.

[10] J. Tao, L. Xu, X. Ma and K. Mei, "An Efficient System for Predicting Hand-Object Contact Probability Based on RGB Image Sequences," 2020 12th International Conference on Advanced Computational Intelligence (ICACI), Dali, China, 2020, pp. 316-321, doi: 10.1109/ICACI49185.2020.9177498.

[11] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," 2019. https://github.com/facebookresearch/detectron2.

[12] Q. Zhao et al., "M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network," arXiv:1811.04533 [cs], Jan. 2019, Accessed: Oct. 28, 2020. [Online]. Available: http://arxiv.org/abs/1811.04533.

[13] R. Chellali and Z. c. Li, "Predicting Arm Movements A Multi-Variate LSTM Based Approach for Human-Robot Hand Clapping Games," 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Nanjing, 2018, pp. 1137-1142, doi: 10.1109/RO-MAN.2018.8525653.

[14] H. Pan and Y. Chen, "Multilevel LSTM for Action Recognition Based on Skeleton Sequence," 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Zhangjiajie, China, 2019, pp. 2218-2223, doi: 10.1109/HPCC/SmartCity/DSS.2019.00308.

[15] C. Vogel, M. Fritzsche and N. Elkmann, "Safe human-robot cooperation with high-payload robots in industrial applications," 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Christchurch, 2016, pp. 529-530, doi: 10.1109/HRI.2016.7451840.

[16] E. Wu and H. Koike, "Real-time Human Motion Forecasting using a RGB Camera," 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 2019, pp. 1575-1577, doi: 10.1109/VR.2019.8798178.

[17] Tekin, Bugra, Federica Bogo, and Marc Pollefeys. "H+ o: Unified egocentric recognition of 3d hand-object poses and interactions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019.

[18] K. M. Kitani, T. Okabe, Y. Sato and A. Sugimoto, "Fast unsupervised ego-action learning for first-person sports videos," CVPR 2011, Providence, RI, 2011, pp. 3241-3248, doi: 10.1109/CVPR.2011.5995406.

[19] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations," arXiv:1704.02463 [cs], Apr. 2018, Accessed: Oct. 30, 2020. [Online]. Available: http://arxiv.org/abs/1704.02463.

[20] Z. Liu et al., "Towards Natural and Accurate Future Motion Prediction of Humans and Animals," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 9996-10004, doi: 10.1109/CVPR.2019.01024.

[21] G. Kapidis, R. Poppe, E. Van Dam, L. Noldus and R. Veltkamp, "Egocentric Hand Track and Object-Based Human Action Recognition," 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, United Kingdom, 2019, pp. 922-929, doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00185.

[22] A. M.V. and D. M. Khan, "Recent Trends on Object Detection and Image Classification: A Review," 2020 International Conference on Computational Performance Evaluation (ComPE), Shillong, India, 2020, pp. 427-435, doi: 10.1109/ComPE49325.2020.9200080.

[23] C. Jian, J. Li and M. Zhang, "LSTM-based dynamic probability continuous hand gesture trajectory recognition," in IET Image Processing, vol. 13, no. 12, pp. 2314-2320, 17 10 2019, doi: 10.1049/iet-ipr.2019.0650.

[24] H. R. Roth et al., "Anatomy-specific classification of medical images using deep convolutional nets," 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI), New York, NY, 2015, pp. 101-104, doi: 10.1109/ISBI.2015.7163826.

[25] L. Lepisto, I. Kunttu, J. Autio, J. Rauhamaa and A. Visa, "Classification of non-homogenous images using classification probability vector," IEEE International Conference on Image Processing 2005, Genova, 2005, pp. I-1173, doi: 10.1109/ICIP.2005.1529965.

[26] F. Pastor et al., "Bayesian and Neural Inference on LSTM-Based Object Recognition From Tactile and Kinesthetic Information," in IEEE Robotics and Automation Letters, vol. 6, no. 1, pp. 231-238, Jan. 2021, doi: 10.1109/LRA.2020.3038377.

[27] N. Coskun and T. Yildirim, "The effects of training algorithms in MLP network on image classification," Proceedings of the International Joint Conference on Neural Networks, 2003., Portland, OR, 2003, pp. 1223-1226 vol.2, doi: 10.1109/IJCNN.2003.1223867.

[28] I. R. Ismaeil, A. Docef, F. Kossentini and R. Ward, "Motion estimation using long-term motion vector prediction," Proceedings DCC'99 Data Compression Conference (Cat. No. PR00096), Snowbird, UT, USA, 1999, pp. 531-, doi: 10.1109/DCC.1999.785688.

[29] D. Jeong, H. Kang, D. Kim and J. Lee, "Mask-RCNN based object segmentation and distance measurement for Robot grasping," 2019 19th International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea (South), 2019, pp. 671-674, doi: 10.23919/ICCAS47443.2019.8971673.

[30] L. Xu, C. Hu, Y. Li, J. Tao, J. Xue and K. Mei, "Deep Conditional Variational Estimation for Depth-Based Hand Poses," 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 2019, pp. 1-7, doi: 10.1109/FG.2019.8756559.