A Trajectory Estimation Approach to a Robotic Air Hockey Opponent

Quinn Leydon
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, NY
qcl4604@rit.edu

Ryan Barry
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, NY
rpb3646@rit.edu

***Abstract*—In this investigation of trajectory prediction, a pivotal component of robotic perception, we concentrate on the realm of air hockey robots. Employing YOLOv8, a cutting-edge single-shot object detection technique, we enhance the detection of the puck's position on an air hockey table. Our approach includes a three-degree-of-freedom planar robot equipped with a flat end effector. Results showcase YOLOv8's outstanding performance in object detection on a tailored dataset and in real-time gameplay scenarios. Despite initial concerns about overfitting, we employ an LSTM model to adeptly capture the non-linearities inherent in games like air hockey, which elude purely velocity-based physics approaches. Real-time metrics underscore the efficiency of YOLOv8 when applied to robotics applications. This research offers valuable insights into the strengths and weaknesses of various trajectory prediction models when integrated into practical robotics applications.**

***Index Terms*—Computer Vision in Robotics, Machine Learning for Robotics, Object Detection, Trajectory Prediction, Air Hockey Robot, Robot Perception**

I. Motivation for Project

The field of trajectory prediction, a pivotal research area in machine vision and machine learning, plays an indispensable role in object tracking and collision avoidance. Its applications span a wide range of areas, including but not limited to, predicting the trajectories of aircrafts, missiles, vehicles, and humans [1]–[4]. The ability to anticipate future positions through prediction is crucial for ensuring seamless and safe path planning, particularly for autonomous agents such as robots or self-driving cars operating within a dynamic environment [5]. Trajectories are commonly modeled as inertial objects influenced by external forces. This paper aims to conduct a comparative study of various endpoint prediction algorithms with respect to the trajectory prediction of a physical object subjected to observable external forces.

II. Literature Review

A. Air Hockey Robots

Air hockey has served as a foundational domain for trajectory prediction research since the late 1990s. In a seminal work, Bishop and Spong employed rudimentary machine vision techniques to predict the endpoint of an air hockey puck. By subtracting sequential frames, they isolated the moving black puck against the white surface. The puck's velocity was inferred from frame-to-frame comparisons. The system continuously updated the current trajectory, accounting for impact locations with walls and minimizing errors. Performance metrics, akin to those used in control systems, included settling time and peak error. Notably, nonlinearities emerged due to bounce characteristics and inconsistent friction from the table surface. Their experimental setup featured a three-link belt-driven robot equipped with a traditional air hockey mallet as an end effector [6].

Namiki, Matsushita, et al. proposed a comprehensive air hockey robot system that integrates short-term and long-term strategies. Leveraging opponent motion and puck position, they employed a high-speed 500 fps camera for tracking. The puck's uniform linear motion was modeled using conventional physics. Short-term tactics involved approaching the impact point, utilizing a 3rd-order differential equation with a 2-link robot. Strategies were categorized into attack, defense, or disregard. Long-term planning included learning opponents' playing styles to inform short-term decisions [7].

In a 2020 study, researchers introduced advanced techniques for an air hockey robot system [8]. Transitioning from frame-by-frame comparisons, they incorporated the tiny YOLOv-3 model, facilitating inferences at an impressive rate of up to 30 frames per second. YOLO, an acronym for "You Only Look Once", is a single-shot multi-box detector renowned for its capacity to perform high-speed detections. This model was employed to detect both the hockey puck and the position of the opponent's mallet. The training phase involved 250 labeled images of the puck in slow motion, and an additional 250 images featuring the puck moving at speeds sufficient to induce motion blur.

Their trajectory prediction was compared to traditional kinematics. The trajectory before collision is given by:

$$X_t = C_x + \frac{Y_t - C_y}{\frac{C_y - P_y}{C_x - P_x}} \tag{1}$$

The reflection law is expressed as:

$$Y_{t+1} = -\frac{C_y - P_y}{C_x - P_x}(X_{t+1} - X_t) + Y_t \tag{2}$$

Here, X_t and Y_t represent the final position before hitting the edge of the table, while C_x and C_y denote the center point

of the puck at the current time. P_x and P_y correspond to the center at the previous moment.

Additionally, after hitting the edge of the table, the point in the defensive area is represented by X_a and Y_a. To account for the hockey puck radius, we have:

$$X_a = Y_b - \sin(\tan^{-1}(m)) \tag{3}$$

$$Y_a = \begin{cases} Y_{min} + \cos(\tan^{-1}(m)), & \text{if } Y_b < Y_{min} \\ Y_{max} - \cos(\tan^{-1}(m)), & \text{if } Y_b > Y_{max} \end{cases} \tag{4}$$

Where X_a and Y_a represent the impact points after correction, X_b and Y_b are the impact points before correction, and Y_{min} and Y_{max} define the upper and lower boundaries. The slope of the current direction of travel is denoted by 'm'.

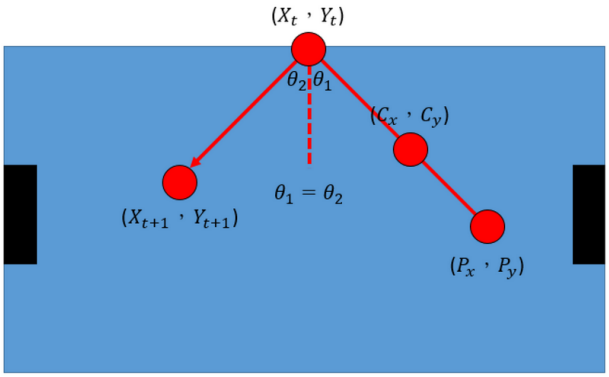

Fig. 1. Schematic diagram of linear formula combined with the law of reflection [8].

In the realm of endpoint prediction, the operational space is partitioned into nine equidistant locations. The task of the neural network is to ascertain the optimal location, among these nine, to which the mallet should be directed. Eight distinct neural network architectures, each characterized by a unique excitation function, were employed with varying degrees of success. The most accurate prediction achieved a commendable accuracy score of 0.80. The robot utilized in these experiments was a one-dimensional entity, propelled by a stepper motor [8].

B. Trajectory Estimation

Highway vehicle prediction using Long Short-Term Memory networks (LSTMs) has been investigated by [9]. Endpoint prediction for highway vehicle tracking is critical for functional autonomous vehicles. These systems must operate at high speeds while maintaining reliability. Missing trajectory predictions can lead to collisions, and abrupt braking uncertainty poses risks to users. The agent leverages local lateral positions and velocities of nearby vehicles. The LSTM model produces a vector of positions k seconds into the future, successfully predicting vehicle lateral positions within a remarkable 70 centimeters at 10 seconds ahead [9].

LSTMs, a subset of recurrent neural networks (RNNs), address the limitations of traditional RNNs. While RNNs maintain information across passes, they suffer from vanishing gradients, hindering their ability to remember long sequences. When memory units are added to RNNs, they encounter exploding gradients due to excessive information storage. LSTMs overcome these challenges through forget gates, allowing them to retain relevant information while maintaining model accuracy [10].

C. Advancements in Object Detection: The YOLO Paradigm

The realm of object detection has been revolutionized by the advent of YOLO (You Only Look Once), a single-shot regression-based model, first introduced in the seminal 2016 paper, "You Only Look Once: Unified, Real-Time Object Detection" [11]. Distinct from the traditional sliding window and template matching methodologies, YOLO employs a single pass through a convolutional neural network for image processing, thereby formulating bounding box estimations within the full image context. This innovative approach significantly mitigates the incidence of false background detections [11].

The YOLO model has undergone more than 20 iterations since its inception, with the most recent versions being YOLOv8 and YOLO-NAS (You Only Look Once - Neural Architecture Search). YOLOv8 represents a significant advancement over its predecessors by incorporating an anchor-free architecture with a decoupled head, thereby enabling the independent processing of objectness, classification, and regression tasks [12].

YOLO-NAS (Neural Architecture Search) further refines this model by integrating quantization-aware modules that pre-parameterize the model for 8-bit quantization, thereby minimizing accuracy loss during this process [12], [13]. Moreover, the inclusion of RepVGG blocks enhances the model's compatibility with post-training quantization tasks. These strategic enhancements not only boost performance on small objects but also augment localization accuracy, rendering the model particularly suitable for real-time deployment on edge devices [14].

III. Methodology

A. Robot

This project introduces a planar robot, equipped with a flat-end effector and offering 3 degrees of freedom (DoF), as depicted in Fig. 2. The robot's design with 270 degree servos at each joint enables comprehensive coverage of an air hockey table's entire hitting zone. Furthermore, it can adjust the orientation of its end effector to redirect the puck toward the adversary's goal effectively.

In this study, we employ a unique kinematic library designed for dynamic adaptation to various robot configurations. This library facilitates the definition of the robot's frames, based on its Denavit-Hartenberg (DH) parameters, and calculates both forward and inverse kinematics of the robot based on its configuration.

Given that the 3 DoF planar robot possesses a readily derivable algebraic inverse kinematic solution, the default iterative method, relying on the Jacobian pseudo-inverse, is replaced by specific inverse kinematic equations tailored to

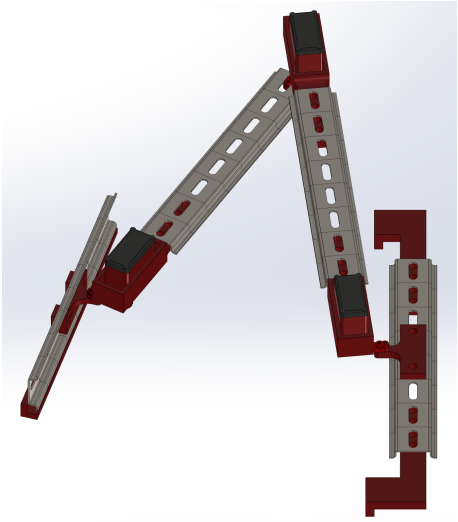

Fig. 2. 3 DoF Robot Arm

the robot's configuration. These inverse kinematic equations, represented from Eq.1 to Eq.5, use (x, y) to indicate the target position and θ_{target} to denote the target orientation of the end effector.

Eq.5 and Eq.6 compute the x and y coordinates of the second joint, addressing a 2 DoF kinematic problem, using the target pose and the length of joint 3. Subsequently, Eq.7 and Eq.8 solve for the angles of joints 1 and 2. Finally, Eq. 9 utilizes these angles to calculate the angle of joint 3. This method effectively demonstrates the versatility and practicality of the library.

$$p_{2x} = x - l_3 \cdot \cos(\theta_{\text{target}}) \tag{5}$$

$$p_{2y} = y - l_3 \cdot \sin(\theta_{\text{target}}) \tag{6}$$

$$\theta_2 = \cos^{-1}\left(\frac{p_{2x}^2 + p_{2y}^2 - l_1^2 - l_2^2}{2 \cdot l_1 \cdot l_2}\right) \tag{7}$$

$$\theta_1 = \text{atan2}(p_{2y}, p_{2x}) - \text{atan2}\left(l_2 \cdot \sin(\theta_2), l_1 + l_2 \cdot \cos(\theta_2)\right) \tag{8}$$

$$\theta_3 = \theta_{\text{target}} - \theta_1 - \theta_2 \tag{9}$$

B. Dataset

An original training dataset was curated through the capture of a 90-minute air hockey session, with still images documented using an OAK-D camera positioned 6 feet above the table surface (see Fig. 3). A comprehensive set of 26,627 images was amassed, and within this dataset, 443 images underwent meticulous hand-labeling with RoboFlow. The table was annotated based on the coordinates of its top left and lower right corners, as illustrated in Fig. 4. Subsequently, these 443 labeled images were utilized to train a YOLOv8 model, which effectively extended its labeling proficiency to the remaining images.

The YOLO model provided bounding boxes for both the table and the puck. To normalize the center position of the puck, Equations 10 and 11 were employed:

$$x_c = \frac{\frac{x_{pr} - x_{pl}}{2} - x_{tl}}{x_{tr} - x_{tl}} \tag{10}$$

$$y_c = \frac{\frac{y_{pr} - y_{pl}}{2} - y_{tl}}{x_{tr} - y_{tl}} \tag{11}$$

Here, x_{pr} denotes the x-coordinate of the puck's right corner in the bounding box, and x_{tl} signifies the upper left corner of the table. Images where the puck was off the table or stationary were excluded, resulting in a refined dataset of 15,763 images. Each image was temporally labeled based on its intersection with the black lines on either the right or left.

To augment the dataset, the table's orientation was mirrored along the vertical axis, doubling the available data to 31,525 samples. Given a slight backward tilt of the table, the dataset was not mirrored along the horizontal axis to preserve this characteristic. This meticulous data curation process laid the foundation for robust model training and evaluation.

Fig. 3. Test Setup

C. Model

The assessment of YOLOv8 and YOLO-NAS models unfolded through rigorous testing on a subset of 443 meticulously annotated images, forming the foundational dataset. Within this carefully curated collection, around 100 images showcased

Fig. 4. Training Image

motion blur, while an additional 64 presented challenges in predicting the puck's position on the table during initial validation. These challenges were particularly pronounced on the table's sides, where the puck was partially obscured by the lip, and over the central tiger region, characterized by its color resemblance to the red puck.

The YOLOv8 model demonstrated far superior performance compared to the YOLO-NAS model, leading to its selection for annotating the remaining training data. The estimation of puck velocity and trajectory involved leveraging essential data points, including the central and edge coordinates of the puck, normalized table coordinates, and the associated timestamps for each image.

An LSTM model was crafted to predict the puck's forthcoming crossing point into the robot's hit zone. The model, featuring 80 LSTM units across two layers, utilized the puck's central coordinates and was labeled based on the anticipated crossing position aligned with the left black line. The model processed sequential data with a length of 10, where the final value corresponded to a point within 20 frames of the puck crossing the goal. The model underwent 400 epochs of training.

This LSTM model was then compared against a physics model outlined in Algorithm 1.

Algorithm 1: Physics Model

Data: x, xp, y, yp
Result: Yc, Θ
$dx \leftarrow x - xp$;
$dy \leftarrow y - yp$;
if $dx \approx 0$ **then**
  $Yc \leftarrow 0.5$;
  $\Theta \leftarrow 0$;
else if $dx > 0.0001$ **then**
  $Xc \leftarrow dx \cdot \frac{(1-y)}{dy} + x$;
  if $Xc > 0$ **then**
    $x \leftarrow Xc$;
    $y \leftarrow 1$;
    $dy \leftarrow -dy$;
else if $dx < -0.0001$ **then**
  $Xc \leftarrow dx \cdot \frac{(0-y)}{dy} + x$;
  if $Xc > 0$ **then**
    $x \leftarrow Xc$;
    $y \leftarrow 0$;
    $dy \leftarrow -dy$;
$Yc \leftarrow dy \cdot \left(-\frac{x}{dx} + y\right)$;
$\Theta \leftarrow \arctan\left(\frac{dy}{dx}\right)$;

IV. Results

A. Puck and Table Detection

1) YOLO-NAS: In the deliberative process leading to the selection of YOLOv8 as our model of choice, an initial experimentation involved the training and testing of a YOLO-NAS model on the foundational dataset, resulting in suboptimal performance after training for 1000 epochs. As evident in Fig. 5, the model exhibited a tendency to inaccurately predict bounding boxes, extending beyond the key points of the table. Notably, it struggled to accurately identify the puck's position on the table, and false detections of the table itself were observed. This discernible behavior aligns with the trends illustrated in the associated confusion matrix, portrayed in Fig. 6.

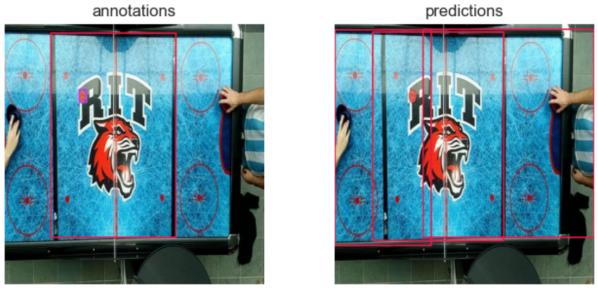

Fig. 5. Validation Prediction by YOLO-NAS Model

2) YOLOv8: A YOLOv8n model, leveraging the foundational dataset and training for 97 epochs before freezing its weights due to early stopping, achieves commendable proficiency in object detection. The precision-recall curve illustrated in Fig. 7 distinctly reveals the model's ability to maintain a notable equilibrium between precision and recall, culminating in a remarkable mean Average Precision (mAP) score of 0.995 at a confidence threshold of 0.5. Furthermore, the F1 confidence curve, as depicted in Fig.8, underscores the model's capability to accurately predict bounding boxes at an exceptionally high rate, even with a low confidence threshold.

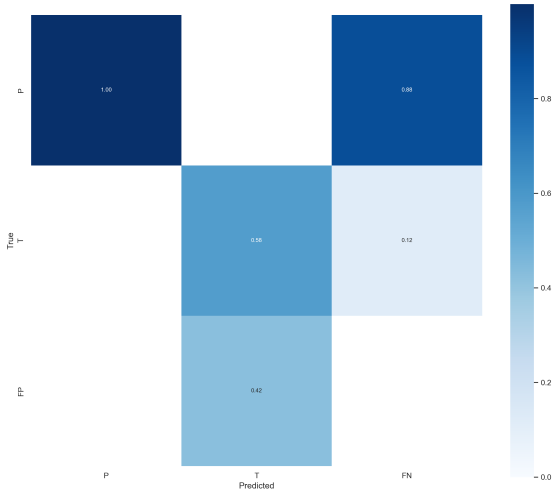

Fig. 6. Confusion Matrix for YOLO-NAS Model

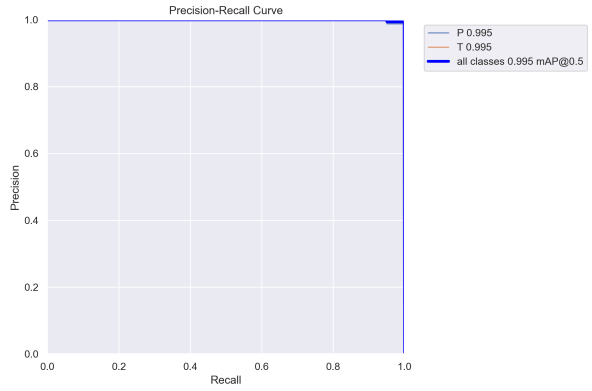

Fig. 7. Confusion Matrix for YOLOv8n Model

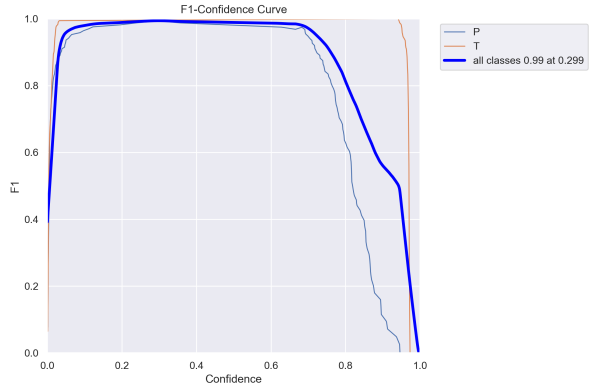

Fig. 8. F1 Confidence Curve for YOLOv8n Model

This observation is reinforced by the impeccably depicted confusion matrix in Fig. 9, further affirming the model's exemplary performance on both the validation and testing datasets.

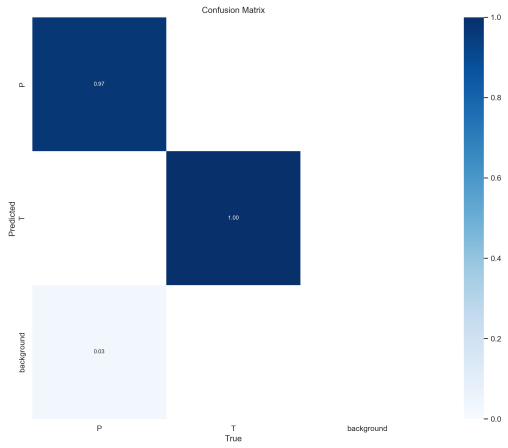

Fig. 9. Confusion Matrix for YOLOv8n Model

The model's precision is notably highlighted in Fig. 10, where predictions made during the training process on a validation batch showcase impeccable accuracy, thereby serving as a compelling testament to the model's exceptional capabilities with an absolute absence of errors. Leveraging the reliability of this highly accurate model, it was employed as a robust data labeling tool, proficiently assigning bounding boxes for key points on the puck and table within the comprehensive dataset comprising 26,627 images.

Fig. 10. YOLOv8n Validation Batch Predictions

B. Prediction Models

The LSTM underwent an extensive training regimen spanning 400 epochs. Despite the incorporation of regularization and weight decay measures, a discernible divergence emerged between the model's training loss, which continued to decrease, and the validation loss, which exhibited an upward trend—clear indicators of overfitting. In response to this, the number of epochs was judiciously reduced to 200 to optimize performance.

The test dataset comprised 2664 sequences, each with a length of 10, encompassing a total of 134 recorded crossings. Evaluation of the predictive performance revealed that the physics model, when applied to sequence data, exhibited a mean squared error (MSE) of 0.1601 for forecasting future crossings, whereas the LSTM achieved a lower MSE of 0.0573. This signifies the LSTM's adeptness in capturing some of the nonlinearities inherent in the table dynamics.

Furthermore, the models were evaluated based on their ability to accurately intercept the puck, defined as predicting the final crossing within a margin of 4.1 inches. In testing data, the physics model successfully intercepted 127 out of 134 shots, while the LSTM achieved a slightly lower success rate, intercepting 109 out of 135 shots. These nuanced results underscore the model-specific nuances in their performance, shedding light on their respective strengths and areas for refinement.

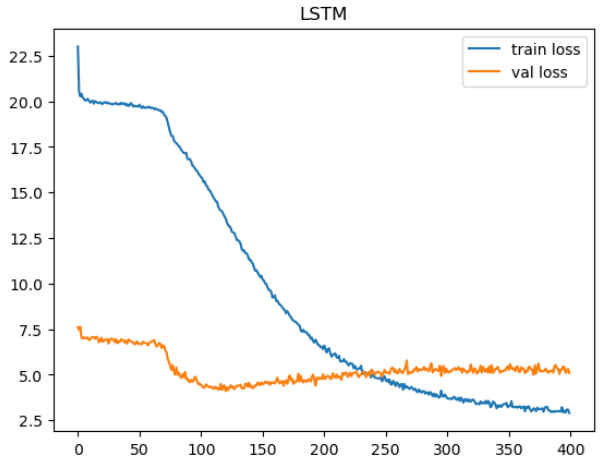

Fig. 11. Validation Curve

C. Real-Time Performance

1) Inference Rate: The OAK-D camera, operating seamlessly at a rate of 60 frames per second (FPS), posed no hindrance to the computational prowess of the system. Leveraging the formidable hardware of an Acer Predator Triton 500 laptop, featuring a 12th-Gen Intel Core i9-12900H processor and an NVIDIA GeForce RTX 3080 Ti Laptop GPU, our integrated system demonstrated exceptional performance. Through the integration of a YOLOv8 model for precise bounding box predictions and a combination of physics-based and LSTM-based approaches for hit zone crossing estimates, our system

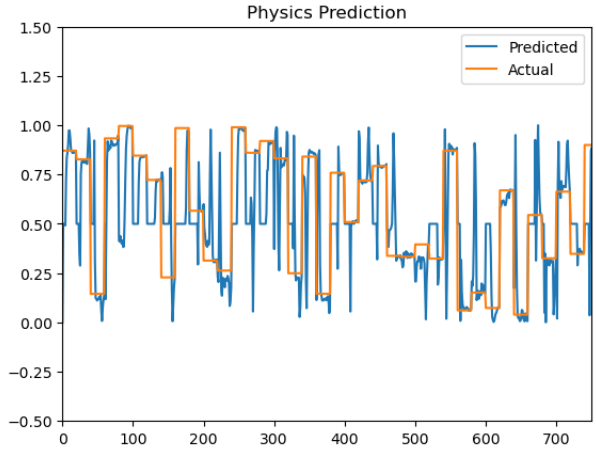

Fig. 12. Normalized Physics Model Location Estimates

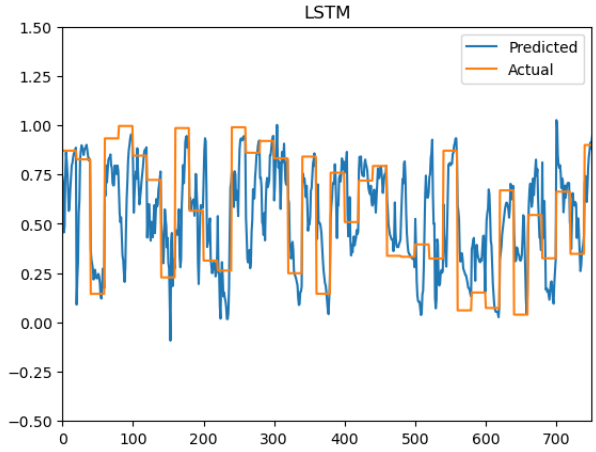

Fig. 13. Normalized LSTM Location Estimates

achieved commendable results. Specifically, we attained a frame rate of 33 FPS while rendering images annotated with bounding boxes. Notably, the system exhibited an even more impressive performance of 57 FPS when not rendering images. This affirms that the primary constraint on the inference rate is indeed dictated by the frame rate of the camera, underscoring the robustness of our integrated approach.

2) Robotic Implementation: A comprehensive lookup table of kinematic solutions, strategically crafted to encompass the robot's hitting zone, was meticulously generated. This approach not only places a paramount emphasis on ensuring the safety of the robot but also serves the dual purpose of mitigating real-time computation demands. By employing a lookup table, the robot adeptly intercepts the puck at its nearest predefined valid kinematic solution without the necessity of dynamically computing the requisite joint angles in real-time. This not only enhances efficiency but also acts as a safeguard, preventing the selection of joint angles that could inadvertently result in collisions with the table's side wall, potentially jeopardizing the integrity of 3D printed joints.

Fig. 14. Normalized LSTM Location Estimates

The robot's three servos were controlled using an SSC-32U PWM servo driver board, enabling precise puck tracking, interception point prediction, joint angle calculation for the interception pose, and direct servo control via serial communication. Fig. 15 visually demonstrates the system's effectiveness in detecting and intercepting the puck, with occasional false detections during testing addressed by concealing the red joints with blue tape.

While the physics model demonstrated proficiency in determining interception positions, consistent with findings in existing literature, the LSTM-based prediction model showed promise in guiding the robot along the puck's trajectory. However, challenges such as vibrational loosening of 3D-printed joints during gameplay limited our ability to fully observe its performance. Recognizing these challenges, future iterations of this project will involve a redesign of the robot's joints to address issues related to wear and tear on the current 3D printed parts, particularly their narrow and closely spaced screw holes. This strategic adjustment aims to enhance the adaptability and efficacy of our robotic system in the dynamic context of air hockey gameplay.

V. Conclusions

This study delves into the critical realm of robotic perception, specifically focusing on the indispensable domain of trajectory prediction. This aspect is foundational in both machine vision and machine learning, finding diverse applications across fields like aircraft, missile guidance, vehicles, and autonomous robotic agents. The ability to accurately predict the future positions of observable objects is paramount for the seamless operation of robotic systems, particularly in stochastic and dynamic environments. In this study, we

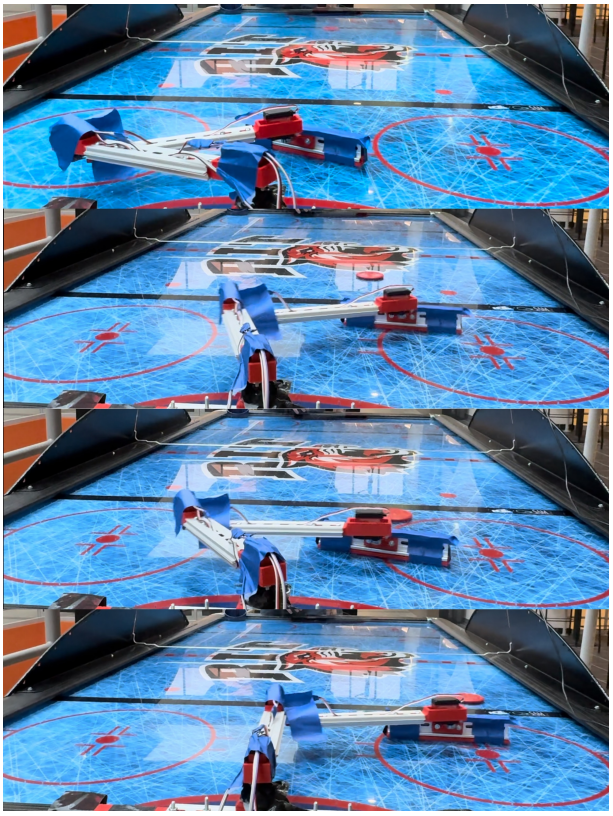

Fig. 15. Robot Hitting Moving Puck

specifically address the challenges of trajectory prediction for a moving object subject to observable external forces, conducting a comparative analysis of various endpoint prediction algorithms.

Our exploration of trajectory prediction in the context of air hockey robots involves tracing the evolution of existing research in this domain. From rudimentary machine vision techniques, as applied in prior works, to more advanced approaches exploring memory-based physics estimation, our design builds upon and extends previous methodologies. Notably, we leverage modern state-of-the-art computer vision techniques, such as YOLOv8 for object detection, and machine learning-based trajectory estimation, departing from purely velocity-based predictions.

Furthermore, our methodology introduces a planar robot with three degrees of freedom, featuring a flat-end effector. This departure from previous approaches, which primarily focused on defending their own goals, sets our implementation apart as it is designed not only to defend but also to actively attempt scoring.

The results of our study highlight a clear distinction in the performance of YOLOv8 and YOLO-NAS when trained for a limited number of epochs. YOLOv8 showcases exceptional proficiency in object detection on our custom dataset. Despite initial indications of overfitting, the LSTM model proves its adeptness in capturing nonlinearities in table dynamics, as evidenced by a lower mean squared error compared to a

physics-based model.

Real-time performance metrics underscore the efficiency of our perception algorithm, achieving high frame rates for detection and tracking, enabling a robust robotic implementation. Leveraging a lookup table for inverse kinematic solutions, our robot effectively intercepts the puck as it enters the hit zone, showcasing the effectiveness of our approach in the dynamic context of an air hockey game.

In summary, our study contributes to the evolving landscape of robotic perception and trajectory prediction, providing insights into the strengths and weaknesses of various models. Moreover, our findings demonstrate the practical integration of these models with both homemade and manufactured robots in real-world applications.

REFERENCES

[1] S. Mondoloni, N. Rozen. "Aircraft trajectory prediction and synchronization for air traffic management applications," in Progress in aerospace sciences, vol. 119, pp. 100640, 2020.

[2] Z. Wang, J. Zhang, W. Wei, "Deep Learning Based Missile Trajectory Prediction," in 2020 3rd International Conference on Unmanned Systems (ICUS), 2020, pp. 474–478.

[3] A. Houenou, P. Bonnifait, V. Cherfaoui and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 2013, pp. 4363-4369, doi: 10.1109/IROS.2013.6696982.

[4] Rudenko, A., et al. "Human motion trajectory prediction: A survey," in The International Journal of Robotics Research, vol. 39, no. 8, pp. 895–935, 2020.

[5] Mangalam, K., et al, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, 2020, pp. 759–776.

[6] B. E. Bishop and M. W. Spong, "Vision-based control of an air hockey playing robot," in IEEE Control Systems Magazine, vol. 19, no. 3, pp. 23-32, June 1999, doi: 10.1109/37.768537.

[7] A. Namiki, S. Matsushita, T. Ozeki and K. Nonami, "Hierarchical processing architecture for an air-hockey robot system," 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 2013, pp. 1187-1192, doi: 10.1109/ICRA.2013.6630722.

[8] Chang, C., et al. "Application of machine learning in air hockey interactive control system," in Sensors, vol. 20, no. 24, pp. 7233, 2020.

[9] F. Altché and A. de La Fortelle, "An LSTM network for highway trajectory prediction," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 2017, pp. 353-359, doi: 10.1109/ITSC.2017.8317913.

[10] K. Smagulova, A. James. "A survey on LSTM memristive neural network architectures and applications," in The European Physical Journal Special Topics, vol. 228, no. 10, pp. 2313–2324, 2019.

[11] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[12] J. Terven and D. Cordova-Esparza, "A Comprehensive Review of YOLO: From YOLOv1 and Beyond," arXiv.org, https://arxiv.org/abs/2304.00501.

[13] X. Chu, L. Li, and B. Zhang, "Make RepVGG Greater Again: A Quantization-Aware Approach," arXiv.org, https://arxiv.org/abs/2212.01593.

[14] M. R. Munawar, "YOLO-NAS vs YOLOv8: A Comprehensive and Visual Analysis of State-Of-The-Art (SOTA) Object Detection Models," VisoByte, https://www.visobyte.com/2023/05/yolo-nas-vs-yolov8-comprehensive-and-visual-analysis-of-latest-object-detection-models.html