Project Golfie

Akkamahadevi Hanni
Department of Computer Science
Arizona State University
ahanni@asu.edu

Andrew Boateng
Department of Computer Engineering
Arizona State University
aoboaten@asu.edu

Kunal Kumar
Department of Computer Science
Arizona State University
kkumar43@asu.edu

***Abstract*—Despite the evolution in robotics, learning robot manipulation task is still an extant challenge in the field of robotics. This is mainly due to the complexity involved in encoding and computing multiple degrees of freedom in the learning process. In this project, we implement a simple golf playing robot which hits a golf ball into a target hole. The idea is to learn the amount of torque to be applied on the golf club based on the distance between the golf ball and the target hole such that the ball enters the hole. The distance between the golf ball and the hole is estimated using object detection and contour area techniques.**

I. Introduction

The field of robotics has evolved into great heights during the last few decades. From performing automated tasks on automotive assembly line to self-driving cars. However, learning complex robot manipulation tasks is still an extant challenge in robotics. [1] gives a good overview about challenges in robot manipulation tasks. Learning robust manipulation tasks come with an overhead of high computation and learning complexity, simply because of the number of degrees of freedom involved in the learning space. Consider a simple task of having a robotic arm (identical to that of the human) pick up a coffee mug. The degrees of freedom in a human arm including the fingers is 27. One can only imagine the complexity of encoding the combination of these degrees of freedom in a manipulation learning task. In this project, we attempt to learn the amount of torque a robotic arm must apply on a golf club to push the ball into a hole. Accordingly, we only consider a single degree of freedom in our manipulation task. Our approach includes three components which we will briefly discuss in sections below. The computer vision component captures an image of the experimental environment that includes the robot with attached golf club, the golf ball and the hole. The key role of this component is to detect the flag at the target hole in the frame and estimate the distance between the robot more specifically the golf ball and the target hole purely using Hue-Saturation-Intensity (HSI) information in the image. The second component is the supervised learning approach that learns the amount of torque to be applied on the golf club such that the ball enters into the hole. The third component is a one Degree of freedom (DOF) manipulator which executes the act of hitting the golf ball with a certain amount of torque in a single direction. The applications of this work can be extended in training a robot in bowling, by increasing the complexity of the model such that it incorporates more degrees of freedom

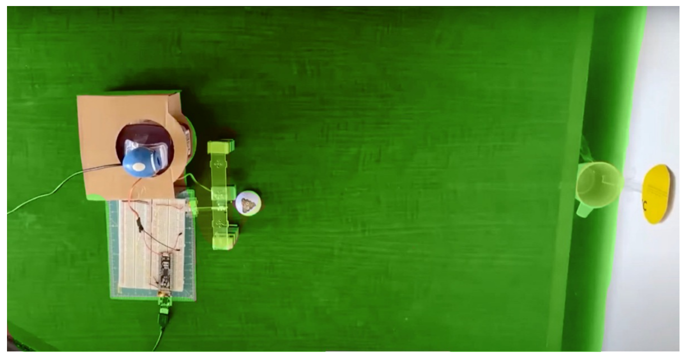

Fig. 1. Experimental setting with 1 DOF manipulator robot with mounted camera, golf ball (white), flag at target hole (yellow)

this can be used to train a robot to play golf in a real golf field.

II. Related Work

Robotics has been evolving since its inception. Its applications are currently used in a lot of different fields such as security, education, sports and entertainment. Apart from soccer, sports is one of the fields that has not seen a lot of use and application of robotics [3]. There has been some work done on robots playing golf such as LDRIC by Golf Laboratories which were covered in the news. Mini-golf playing robots however were only created to help teach students and robots [4]. [5] has a very good implementation of playing mini-golf, the authors focus on ways to actually put the ball into the hole and figure out the angles and powers needed to make a put. The difference between our project and [5] is that, our project uses machine learning to determine the distance and the torque needed while their work focuses on inverse kinematics and fuzzy logic control of the robot. Also, our work focuses on estimating the distance between the ball and the hole accurately and determining the torque needed to hit the golf ball into the hole.

III. Methodology

For the sake of simplicity, in this project, we set up our environment in such a way that the golf ball, the golf club and the target hole are all linearly arranged. Therefore, we establish that when the golf club nudges/hits the ball, it rolls down linearly in the direction of the hole. We achieved this by restricting the motion of the club with a single degree of freedom. We have three components in our project, a) Computer vision component: to estimate distances, b) Learning component: to learn the amount of torque to be applied and the success/failure of a hit and c) Robot Manipulation component: to execute the process of hitting the golf ball into the hole.

A. Computer Vision

We use Computer Vision for two aspects: detecting the target hole in the frame and estimate the distance between the camera (mounted on the robot) and the hole. We use a concept of computing contour area for estimating the distance between the robot and the hole. Contour is the boundary of the flag (placed near the target hole) which we want to use to estimate the distance to hole. The idea behind this is that the object appears bigger when it is closer to the camera and smaller when it is farther away from the camera. We used a method called ContourArea() from the OpenCV library which calculates the area of the pixels enclosed by the object. For applying the contour, we first use the threshold detection or canny-edge detection techniques which helps in the masking of the object from its boundaries. Essentially, it looks for a significant difference in the magnitude of the pixel values and tries to find a recognisable shape, in our case a circular area and draws a boundary around it. Once we have masked everything else in the image apart from the circular boundary and the pixels enclosed in it, we can use this to estimate the area of the circular flag. We use the parabolic equation between the hole and the area of the flag to predict/estimate the distance to the hole. We will discuss how we obtain this equation in the experimental evaluation section.

B. Learning Approach

Given that the golf club, the ball and the target hole are all linearly aligned, our aim is to learn the minimum amount of torque required to hit the ball such that it enters the target hole. The features we use for our learning are 1. The estimated distance between the robot and the target hole return from the computer vision component and 2. The discretized torque level applied on the golf club. The corresponding labels indicate whether the hit resulted in success or failure.
We use two supervised learning approaches to learn the torque to be applied on the golf club. The first algorithm we use is a neural network with stochastic forward passes where we build a Multi-Layer perceptron (MLP) with drop out layers using keras libraries. The second algorithm we use is the K-Nearest-Neighbor (KNN) algorithm. The reason behind choosing KNN is that it is very suitable for minimal features and a small dataset and we will see in the results section that it works extremely well for this task.

C. Robot Manipulation

The computer vision component estimates the distance between the robot and the target hole and provides this as an input to the learning model. The learning model predicts the result of a hit with the estimated distance as an input along with the set of discretized torque levels as a vector and obtains a prediction for all the torque levels. We further choose the minimum torque value which has a prediction as success among all of levels of torque. This torque value is sent to our 1 DOF manipulator robot using UART comumnicatiom. The robot executes the hit with the input torque value sent to it.

IV. Experimental Evaluation

The experiments are setup on a table with a cup attached to the end of a table which makes up the hole or "the cup" of the golf course as shown in Fig. 1. In addition, we mount a yellow circular shaped flag unlike an actual golf pin flag over the golf hole. For the hardware of the golf player robot, We built a simple 1 DOF manipulator with a servo attached to plastic bars. The end-effector (golf club) was made of a flat plastic and the legs were made with Lego's. Note: We initially planned of 3D-printing a human-like robot but could not, due to the closure of all school labs as a result of the COVID-19. We used a servo instead of a DC motor because servos have unique arrangement which allows the motor to rotate at a specific angle with greater accuracy and precision. Also, servos use feedback system so it takes corrective measures when we are far away from the desired output. The servo used in this project is the micro servo SC90 by Tower Pro. Our programming chip is the PSOC CY8C5888LTI-LP097 by Cypress. We chose this particular chip due to its big online support, built in support for PWM, familiarity and simple to use IDE called the PSOC creator.

The main variables we worked with were the distance to the hole from the camera and the delay value between every one degree of an angle. We learned that the delay value is inversely proportional to the torque applied by the servo. The higher the delay value, the slower the servo turns and by association, lower is the torque value applied on the end-effector (golf club). A Fast moving end-effector means the ball is going to get hit with a higher torque.

The PSOC IDE does not allow us to directly set angles for the rotation of the servo so we had to do the come up with a formula to convert the "compare value" to angles. We know that the servo moves to the most anticlockwise when the compare value is set at 1500 and it goes the most clockwise when the compare values are set at 6900. Also, servo's have 180 degree rotation power. The relation between the servo angles and the compare value is as shown in Fig. 2 is a linear relation. We determine the resulting equation by using the 2-point form of linear equation.

To detect the actual distance between the robot and the hole, first we train a model by plotting a graph between the manually measured distance and the area of the flag at the hole based on an image captured from the camera. For the purpose of easy and accurate recognition, we use a yellow colored identifier seen in Fig. 1 next to the target hole for estimating the distance to the hole as shown in Fig. 3. We mask the entire image apart from the recognized 'yellow' region. The resulting image is used to compute the area of the flag and simultaneously record the actual distance. We move the robot to different locations in the space and manually record the distance between the

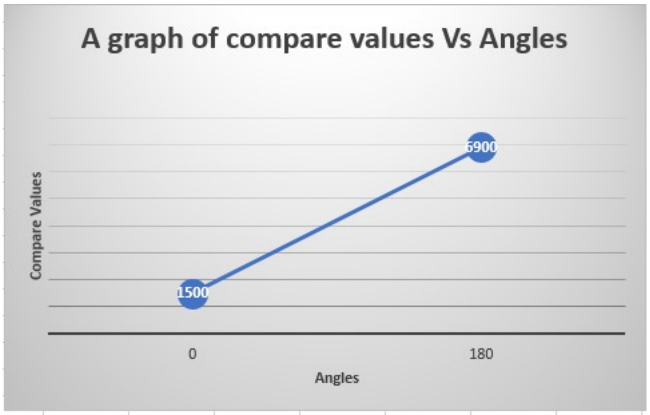

Fig. 2. Relation between compare values and angles of the servo

Fig. 3. Distance estimation based on contour area computation. Green border indicates the detected 'yellow' region in the image. Blue border indicates the approximated circular region of the flag.

hole and the robot and corresponding area computed by the ContourArea() function is also recorded. After this step, we plot a graph and fit a parabolic equation for the calibration data points of distance measured and area of the contour as shown in Fig. 4. Therefore, we get a quadratic equation in x and y where x denotes the area of the contour and y denotes the distance to the hole.

We experimented with two different learning models 1. MLP with stochastic forward passes and 1. KNN algorithm. The MLP consists of an input layer with 32 nodes and two hidden layers with 16 and 8 nodes respectively. We add dropout layers in between to ensure there is no over fitting and to simulate stochastic forward passes. This model is trained with Adam and RMSProp optimizers for around 50 epochs. The second model i.e. KNN is implemented with K=5. For collecting the training data we record 3 observations of golf hits for each torque level at each position of the robot. We do so for around 8 different positions on the table and obtain around 200 data samples.

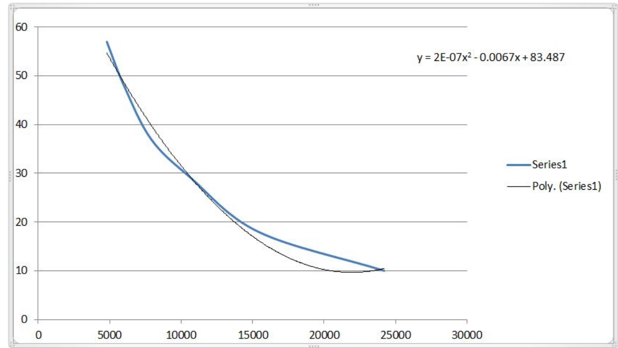

Fig. 4. Parabolic equation obtained upon fitting the calibration data points

To summarize the flow of work, the camera captures an image and estimates the distance d to the hole. The torque to be applied is discretized into 5 different levels in our experiments, where level 1 corresponds to the highest amount of torque and level 5 corresponds to the lowest amount of torque. The distance d transformed to a vector of length 5 and is combined with a vector of levels of torque [1, 2, 3, 4, 5] to form an input vector of size 5×2. The first column indicating the distance to hole (feature 1) and the second column indicating the torque level (feature 2). We obtain the predictions from the model for each torque level and we choose the minimum level torque that has a prediction as success. This value is sent to the servo using UART and the servo executes the hit with the corresponding delay and we record the result as one observation.

V. Results and Analysis

The distance estimation was very accurate with an error rate of less than 5% for each data sample. For example, the distance estimated for the given image in Fig. 3 is 19.196 while the measured distance for the same instance is 19.0. Therefore, we can say that the distance feature is sound and reliable.

We used two different machine learning models to analyze which one works best for our project. We initially modelled our learning using a Neural Network (MLP) with stochastic forward passes. The training and validation accuracy obtained for this model was very low (around 51%). We believe that the reason behind this is that neural networks require a large amount of data samples as opposed to our small dataset. We switched to KNN algorithm and obtained a validation accuracy of 79%. For testing our project, we performed a total of 11 trials using the KNN algorithm and observed that 9/10 hits resulted in a successful goal.

The videos for training can be found [here] or go to the url: https://youtu.be/IC3q3gWFIgc

The videos for testing can be found [here] or go to the url: https://youtu.be/Ge0AcEQolOU

VI. Conclusion

This project gave us an insight to simplicity of solving a problem when the number of degrees of freedom is low. As the number of degrees of freedom increases the complexity of the problem increases exponentially. Complex models like MLP does not perform well with small datasets and at some point we observed that adding dropout layers further hampered the performance of the mode. Hence using a KNN model for learning such a simplistic task was the right way to go. Another interesting observation we made was that the trajectory of the ball was different for the exact position, distance and level of torque. This was a strange behaviour and made us wonder if there's something wrong with our experiments. Eventually we realized that the stochasticity in trajectories was due to the uneven texture of the golf ball. This was a valuable insight to the fact that simulated environments no matter how well designed cannot equate a real environment.

References

[1] C. C. Kemp, A. Edsinger and E. Torres-Jara, "Challenges for robot manipulation in human environments [Grand Challenges of Robotics]," in IEEE Robotics Automation Magazine, vol. 14, no. 1, pp. 20-29, March 2007.

[2] S. Mahamud and M. Hebert, "The optimal distance measure for object detection," 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., Madison, WI, USA, 2003, pp. I-I.

[3] J.H.Kim, D.H.Kim, Soccer Robotics, Springer, 2004, ISBN: 3540218599

[4] Jouaneh, M. Carnevale, P., "Autonomous Robotic System for Playing Mini-Golf", IEEE Robotics and Automation Magazine, Courtesy of The Center for Robotics Automation Research, University of Rhode Island, June 2003.

[5] Jabson, N Leong, K Licarte, S Oblepias, G Palomado, E Dadios, Elmer. (2008). "The autonomous golf playing micro robot: with global vision and fuzzy logic controller". International Journal on Smart Sensing and Intelligent Systems. 1. 10.21307/ijssis-2017-323.