10. Project work: Beer pong

Simon Pfingstl, Machine Learning for Product Development, WiSe 2025/26

1. **Test setup**
   Build up a test setup with the following equipment:
   a) A table about 2.4m long and 0.75m wide (very roughly) serving as a beer pong table.
   b) A big piece of paper (60cm x 60cm) with a printed/drawn coordinate system and grid, which serves as your target. The paper sheet is placed on one side of the table and is used to measure where the ping-pong ball lands. The center of the coordinate system is to be placed in the middle of the table, according to its width, and 30cm measured from the end of the table (length).
   c) Two cameras (such as smartphones, webcams, etc.) are to be mounted, e.g., on camera stands around the table. Ensure you can specify the frames per second (fps) for the cameras. You should be able to set the same fps for both cameras. Use at least 60fps. This is typically possible with any smartphone.
   d) Take a picture of your test setup and specify the important distances (table, target, etc.).
2. **Collecting data**
   Record videos of you throwing the ball towards the target (piece of paper) and determine its landing position.
   a) Specify a meaningful camera positioning for the two cameras. A camera positioning specifies the location of the two cameras relative to the table.
   b) For your specified camera positioning, record at least 30 videos synchronously with both cameras and note the landing position of the ball for each throw (delta x and y in cm, measured from the center of the target; positive and negative values).
   c) Voluntary: Try to find suitable image data for ping-pong object detection online.
3. **Object detection model**
   Train an object detection model to detect ping-pong balls.
   a) Determine two different applicable object detection models/versions.
   b) Label your data and train the two object detection models using the recorded videos. You might also use data found online.
   c) Compare your trained object detection models based on suitable performance measures. Which one do you use?
   d) Write a Python function with two images as inputs and two centers, widths, and heights of the bounding boxes in pixel coordinates as outputs.
4. **Location tracking**
   Write a code that determines the current location of the ping-pong ball.
   a) Specify a physically meaningful coordinate system for the ball trajectory.
   b) Write a Python function that uses the object detection output to determine the x, y, and z location of the ball.
   c) Visualize the trajectories in multiple different plots, e.g., x, y, z; x(t); y(t); z(t).
   d) Do the inferred/tracked landing points correspond to the measured ones? Quantify the error.
   e) Fit x(t), y(t), and z(t) based on linear regression and quantify the uncertainty of your location prediction. How does the uncertainty evolve over the trajectory, i.e., is your prediction more precise in certain regions/times? Quantify it.

5. **Speed tracking**
   Write a code that determines the current speed of the ping-pong ball.
   a) Use the same physical coordinate system as specified in 4a). Determine the velocity in x, y, and z based on the fitted function in 4e) and use it as ground truth for this task.
   b) Write a Python function that determines the current x, y, and z speed of the ball.
   c) Quantify the error of your speed tracking algorithm.
   d) How can you tackle large uncertainties? List two of your best ideas and show their effect on the uncertainty for speed tracking.

6. **Physics-based prediction**
   Write a code function using a physics equation to predict the future trajectory of the ball.
   a) Which physics equation do you use to predict the trajectory of the ball? Write a code function that predicts the landing point of the ball based on the current location and speed as inputs.
   b) Evaluate the prediction error of the landing point based on your gathered target data, i.e., measured distance from the center of your piece of paper. How does the error evolve over time/the trajectory?
   c) How can you account for the estimated error in location and speed tracking? Try out one idea and show the improvement.

7. **ML-based prediction**
   Write a code function that predicts the trajectory or landing point of the ball based on ML models.
   a) Think of two ML models and describe how you use them in order to predict the future trajectory or landing point of the ball. You might also use the estimated error of the location and speed tracking as inputs.
   b) Implement the two ideas and predict the ball's trajectory and/or landing point.
   c) Compare both models based on the prediction error of the landing point. Use the gathered target data to compute the prediction error.

8. **Conclusions**
   Summarize your results and draw conclusions from your work. Try to answer the following questions:
   a) What are the reasons for your camera positioning?
   b) Which object detection model is the most useful one? Why?
   c) Is location tracking more precise in certain regions than in others? If so, why?
   d) How could you improve the speed tracking? What did not work?
   e) Would you rather use a physics-based or an ML-based landing point prediction? Why?
   f) What was the most challenging task in this project work?
   g) What would be necessary in order to predict the landing point as accurately, such that a real-time setup could move a cup to the landing point of the ball? Which is the most critical step that would have to be improved?

Additional information

- Exercises 1, 2, and 8 are team exercises and should be solved together.
- Exercises 3-7 are personal exercises. Assign them to individual members of your team. You can and should still help each other.
- Prepare slides in order to present the results of all tasks in a 15-minute presentation.
- Your presentation mustn't take longer than 18 minutes.

- The slides should follow the sequence of the exercises. Write down the team member's name on the slides associated with the individual exercise.
- Please use an appendix to include more information about your individual exercises. For each individual exercise, you are allowed to include three appendix slides in your presentation. You most likely won't present these slides. Yet, they are considered for grading your work.
- Upload your presentation, codes, and recorded data on Moodle as one zip file until January 21st. Make sure that the codes are properly named and commented. Furthermore, pay attention to use a proper data folder structure for training a YOLO model.
- The clarity of your documents is also taken into account for your grade.