Boxes and Balls: Final Presentation

—

Owen Huang and Richie Zachariah

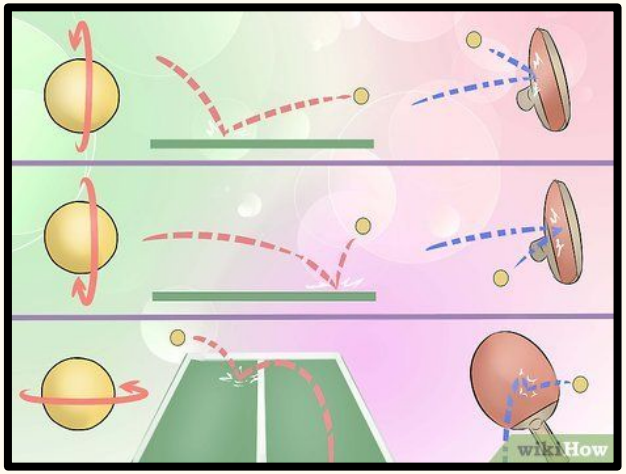
Agenda

1. Background
2. Data Generation
3. YOLO example
4. Spin Detection
5. Next steps

Background

Background: Spin in Table Tennis

- Very dynamic sport
- Players utilize the concept of spin to gain an advantage
- Spin caused by friction between ball and paddle
- Spin:
  - Changes path of ball
  - Changes how opponent must return ball



Background: Our Problem

We wanted to develop an algorithm that can predict what spin was applied to a ping pong ball after it has been hit.

Tasks:

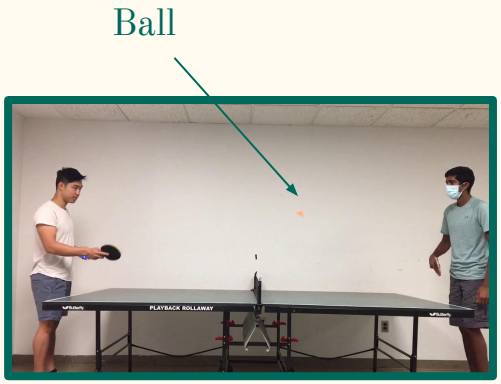
- Train a neural network to recognize a ping pong ball
- Track ball using video
- Follow path of ball
- Classify the spin of the ball based on trajectory
    - Ex. top spin, back spin, side spin

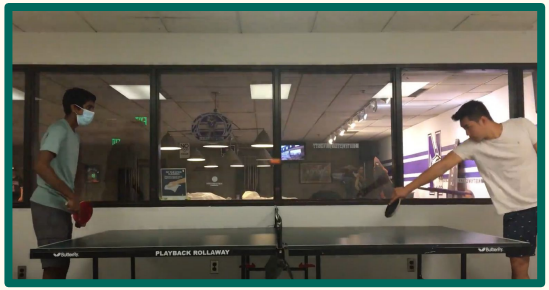
Background: Project Applications

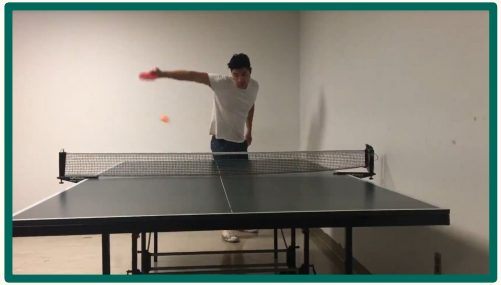
- Gathering statistics and information (Ex. Frequency of certain spins)
- Improving quality of replays
- Analyzing player playstyle
- Similar concept can be replicated in other sports
    - Tennis
    - Soccer
    - Pool
    - Baseball
    - Cricket

Data Generation

Data Generation: Recording Video



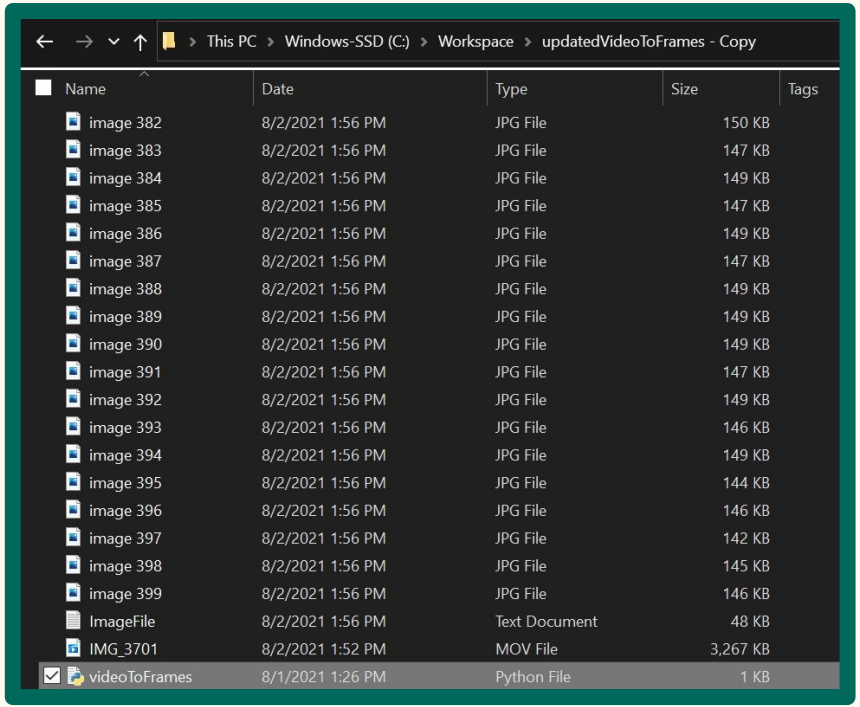




- iPhone camera
- Using 1080p at 60 FPS
- Around 20 minutes of footage
- Used different angles with varying distances

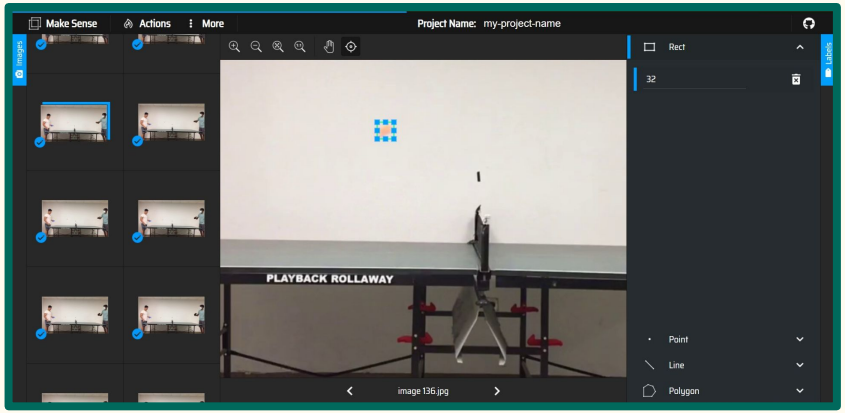
Data Generation: Extracting Frames from Video

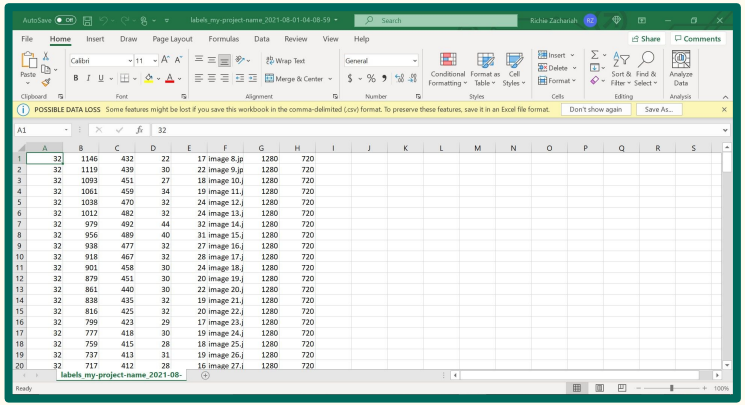
```python
import cv2
import os
import glob

vidcap = cv2.VideoCapture('IMG_3701.MOV')
success,image = vidcap.read()
count = 0

# Saves each frame in same directory as program
while success:
  cv2.imwrite("image %d.jpg" % count, image)     # save frame as JPEG file
  success,image = vidcap.read()
  print('Read a new frame: ', success)
  count += 1

# Creates text file with image numbers and directories
imageFile = open("ImageFile.txt", "w")
count2 = 0
for directory in glob.glob("C:/Workspace/updatedVideoToFrames/*.jpg"):
  # currentImage = os.listdir(name)
  str1 = ("image_%d.jpg, " % (count2))
  str1 = str1 + directory
  str1 = str1 + "\n"
  imageFile.write(str1)
  count2 += 1
imageFile.close()
```



- Converted video to individual frames
- Saved frames as JPG files in folder
- About 60 pictures per second of video

Data Generation: Forming Training Data





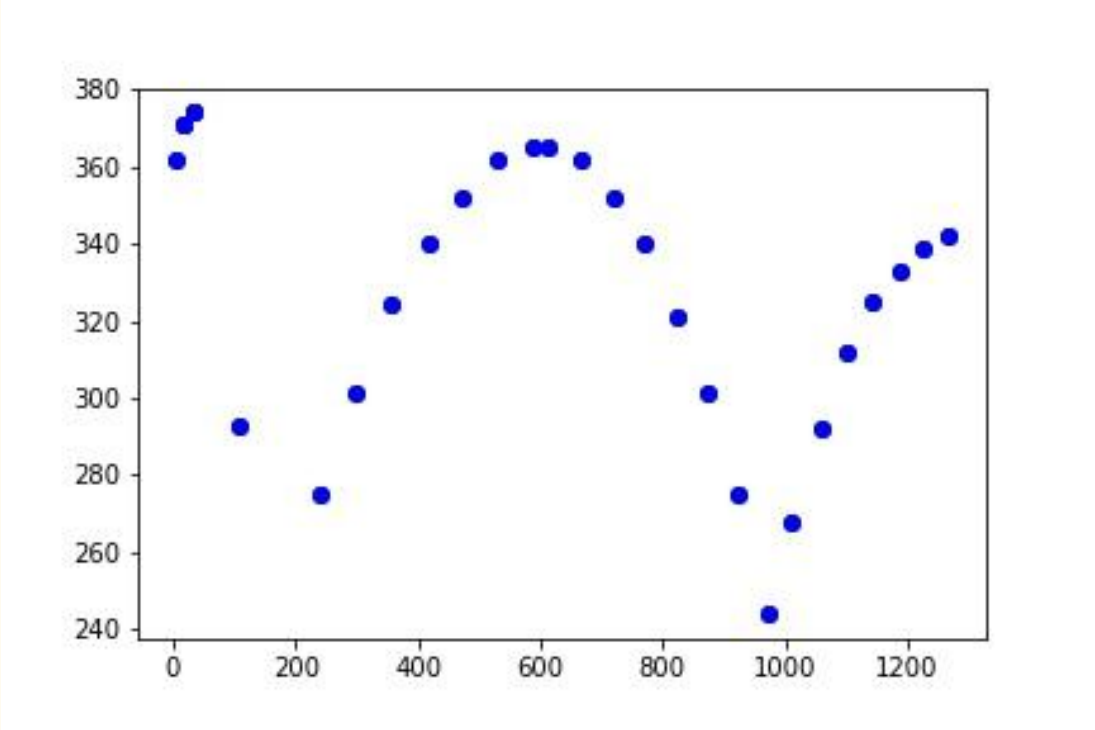
	A	B	C	D	E	F	G	H
1	32	1146	432	22		17 image 8.jp	1280	720
2	32	1119	439	30		22 image 9.jp	1280	720
3	32	1093	451	27		18 image 10.j	1280	720
4	32	1061	459	34		19 image 11.j	1280	720
5	32	1038	470	32		24 image 12.j	1280	720
6	32	1012	482	32		24 image 13.j	1280	720
7	32	979	492	44		32 image 14.j	1280	720
8	32	956	489	40		31 image 15.j	1280	720
9	32	938	477	32		27 image 16.j	1280	720
10	32	918	467	32		28 image 17.j	1280	720
11	32	901	458	30		24 image 18.j	1280	720
12	32	879	451	30		20 image 19.j	1280	720
13	32	861	440	30		22 image 20.j	1280	720
14	32	838	435	32		19 image 21.j	1280	720
15	32	816	425	32		20 image 22.j	1280	720
16	32	799	423	29		17 image 23.j	1280	720
17	32	777	418	30		19 image 24.j	1280	720
18	32	759	415	28		18 image 25.j	1280	720
19	32	737	413	31		19 image 26.j	1280	720
20	32	717	412	28		16 image 27.j	1280	720

https://www.makesense.ai

- Manually placed bounding boxes around ball
- Exported boxes in .csv file
  - X-coordinate of center, Y-coordinate of center, width of box, height of box
- Values above used for machine learning

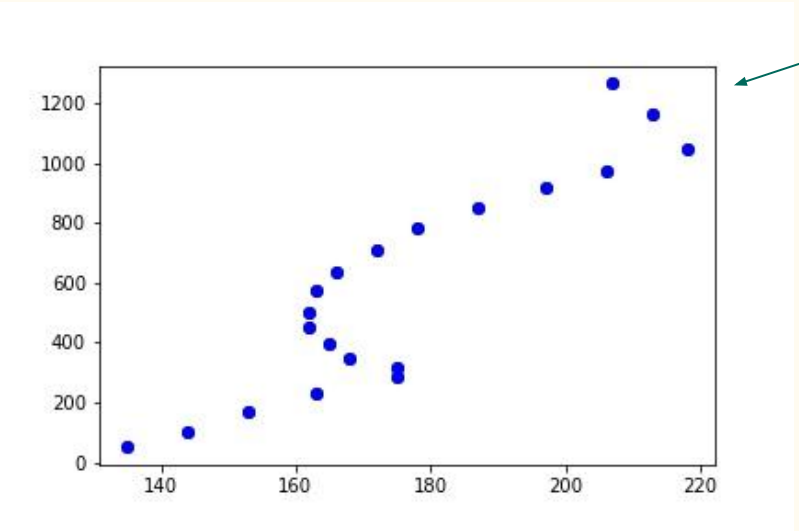
YOLO

YOLO: Example



Spin Detection

Spin Detection: Theory



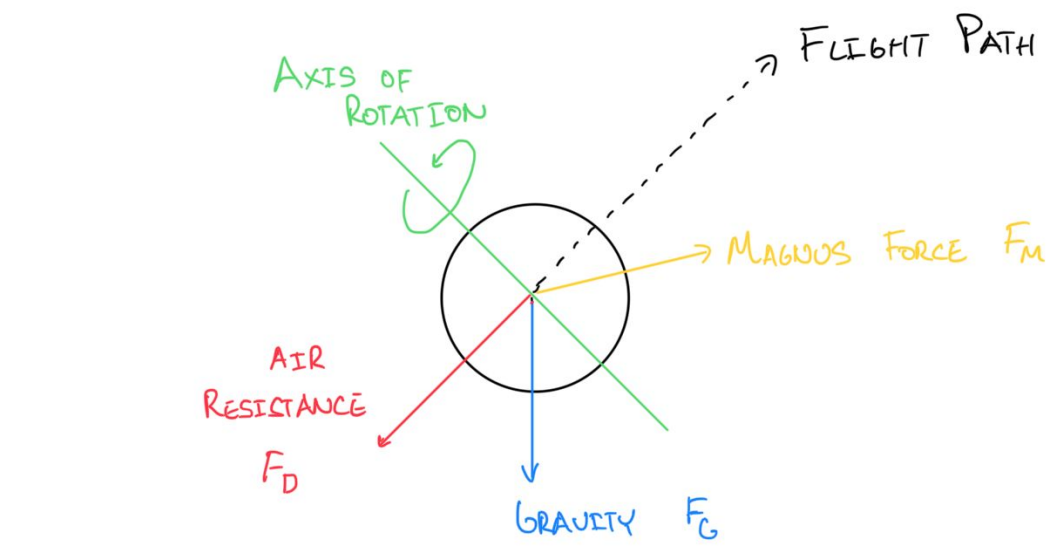
Ball Trajectory

- Exploits flight path and Magnus Force
- Will use polynomial regression

Spin Detection: Extracting 3D Coordinates

- Very crude approximation (because of time constraints)
- Extracting 3d coordinates from 2 camera angles

Spin Detection: Magnus Force



Spin Detection: Regression and Linear Systems

Spin Detection: Results

```
def solveOmega():
    M = buildMatrix(0)
    acc = buildAcceleration(0)
    for i in range(1, len(times)):
        t = time_step * i
        M = np.concatenate((M,buildMatrix(t)),axis=0)
        acc = np.concatenate((acc,buildAcceleration(t)),axis=0)
    omega = np.linalg.solve(M.T@M, M.T@acc)
    print('Ball is spinning in the direction of ' + str(omega))

solveOmega()
```

```
Ball is spinning in the direction of [ 4.80489623e+06  5.58332166e+09 -2.92713315e+06]
```

Next Steps

Next Steps

- Program that can trace 2D path of ball from different angles
- Done using center of bounding box for each frame
- 2-d path important for detecting spin

Questions?