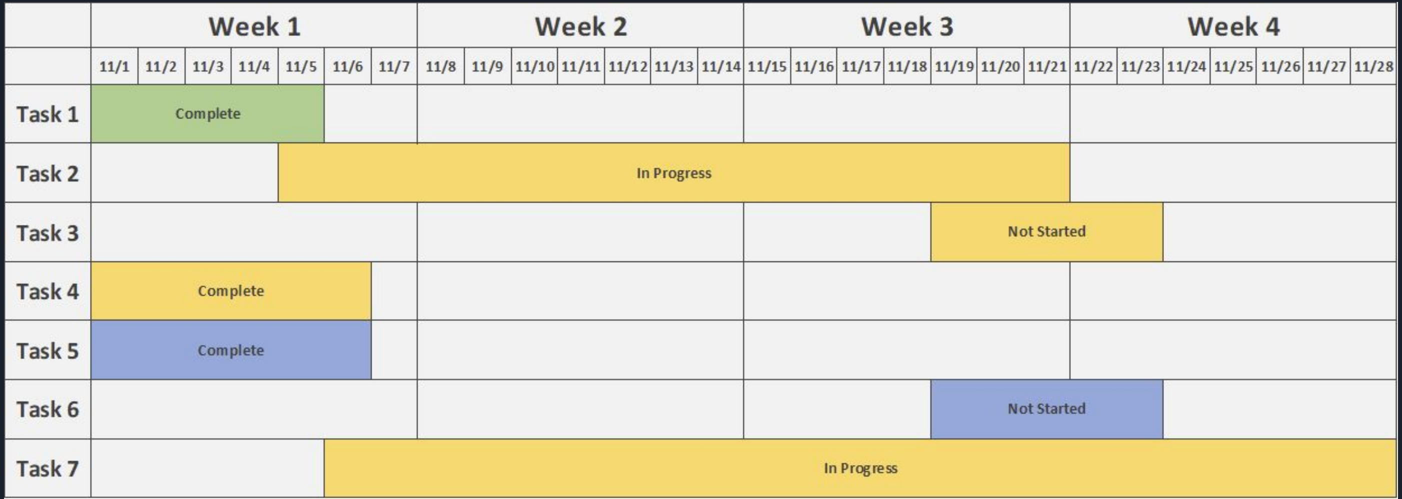
Project 3: Face Detection, Tracking, and Recognition Intermediate Results

Group 6: Eric Watson & Hansi Zheng

Schedule & Roles



	Week 1							Week 2							Week 3							Week 4						
	11/1	11/2	11/3	11/4	11/5	11/6	11/7	11/8	11/9	11/10	11/11	11/12	11/13	11/14	11/15	11/16	11/17	11/18	11/19	11/20	11/21	11/22	11/23	11/24	11/25	11/26	11/27	11/28
Task 1	🟩 Complete	🟩	🟩	🟩	🟩																							
Task 2					🟨 In Progress	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨							
Task 3																			🟨 Not Started	🟨	🟨	🟨	🟨					
Task 4	🟨 Complete	🟨	🟨	🟨	🟨	🟨																						
Task 5	🟦 Complete	🟦	🟦	🟦	🟦	🟦																						
Task 6																			🟦 Not Started	🟦	🟦	🟦	🟦					
Task 7						🟨 In Progress	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨

Tasks

1. Train YOLOv5 Model on Face
2. Implement a system for detection, tracking, recognition
3. Design live demo for Zoom
4. Determine camera FOV and angular resolution
5. Determine max distance from camera for system operation
6. Record system resource usage
7. Work on final presentation

Roles

Blue - Hansi Green - Eric Yellow - Both

Camera Details (Logitech C920/X)

- Max resolution of 1080p @ 30 FPS
- Lens focal length = 3.67 mm
- Lens diameter = 2 mm
- Angular resolution (Δθ)
    - White (880 nm) = 536.8 μrad
    - Red (660 nm) = 402.6 μrad
    - Green (520 nm) = 317.2 μrad
    - Blue (470 nm) = 286.7 μrad
- Horizontal FOV = 70.42°
- Vertical FOV = 43.3°

Logitech C920

Face Model

- For face detection, the YOLOv5 library [1] is used, with the V5S model being chosen as the face detector.
- For face tracking, the DeepSort library [2] is used with the YOLOv5 library.
    - Compared to using a Kalman Filter for tracking, DeepSort would allow for tracking a face with occlusions/obstructions.
- For face recognition, the Face Recognition library [3] was initially chosen.
    - Impacts inference time when encoding detected faces, so another library or method is needed.
    - Took approximately 2 seconds per frame to encode the detected face.

Face Dataset & Training

- The Open Images Dataset [4] was used train the YOLO V5S model.
  - 40,000 samples of the ‘Human Face’ class were used.
  - 30% of the samples were used as the validation set.
  - Annotations were converted to format used in YOLOv5.
- The YOLO V5S model was trained with the following parameters:
  - Epochs: 100
  - Batch Size: 16
  - Image Size: 640
  - Initial Learning Rate: 0.0032
  - Momentum: 0.843
  - Weight Decay: 0.00036
  - IoU Threshold: 0.2

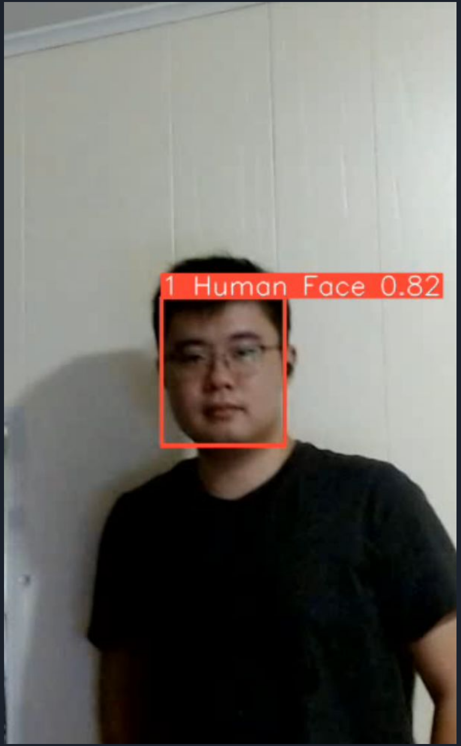
Face Detection & Tracking Results

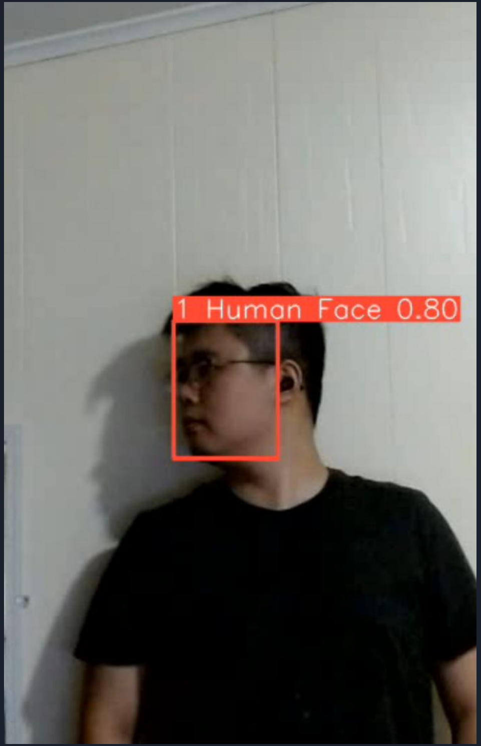
- Video Resolution = 1920 x 1080 @ 30 FPS
- Detection/Tracking Resolution = 640 x 384
- Detection/Tracking Inference = 0.039 seconds (25.64 FPS)

Confidence Based on Distance

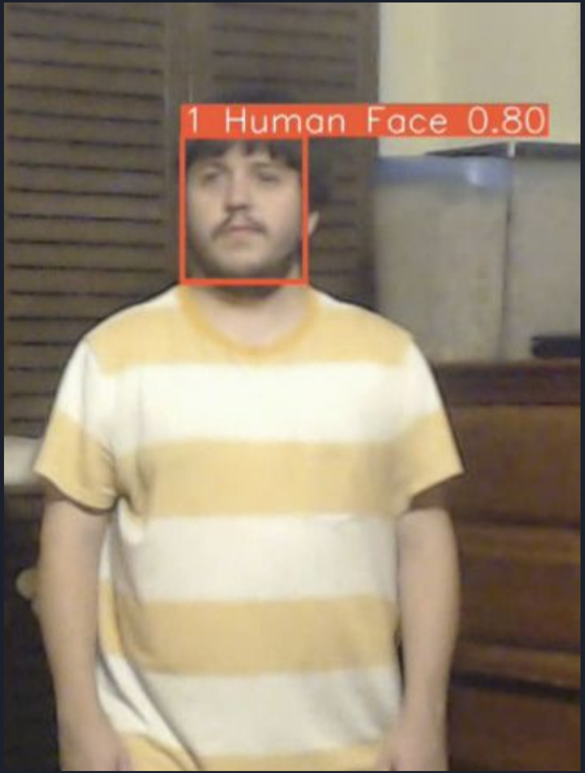
	Close (2 FT)	Medium (6 FT)	Far (9 FT)
Front (Hansi)	0.82 ~ 0.92	0.81 ~ 0.86	0.79 ~ 0.84
Front (Eric)	0.86 ~ 0.90	0.83 ~ 0.86	0.80 ~ 0.85
Side (Hansi)	0.84 ~ 0.91	0.80 ~ 0.83	0.77 ~ 0.85
Side (Eric)	0.72 ~ 0.86	0.73 ~ 0.80	0.75 ~ 0.80

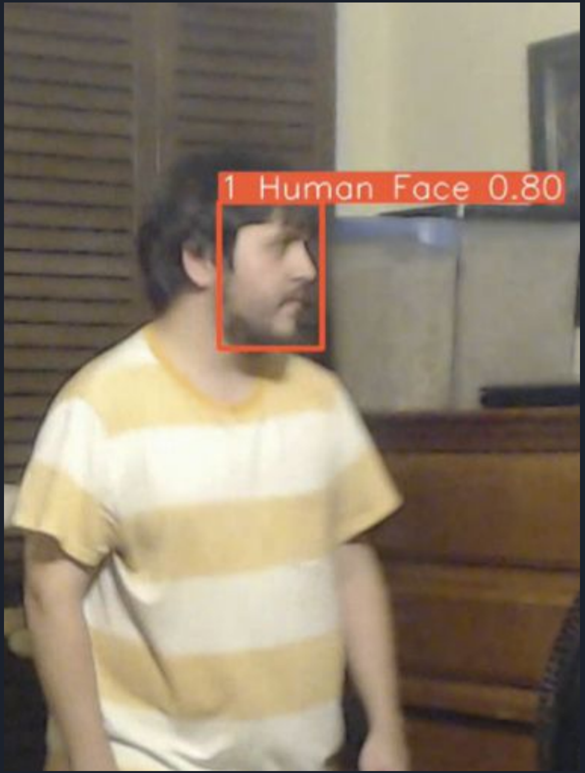
Face Detection & Tracking Results (Hansi)





Face Detection & Tracking Results (Eric)





Remaining Tasks

- Find a better method to implement face recognition.
- Finish implementing the system for face detection, tracking, and recognition.
- Design a live demo for Zoom.
- Record power and temperature of the system when operating.
- Work on final presentation.

References

1) YOLOv5 Library

   https://github.com/ultralytics/yolov5

2) YOLOv5 + DeepSort

   https://github.com/mikel-brostrom/Yolov5_DeepSort_Pytorch

3) Face Recognition

   https://github.com/ageitgey/face_recognition

4) Open Images Dataset

   https://storage.googleapis.com/openimages/web/index.html