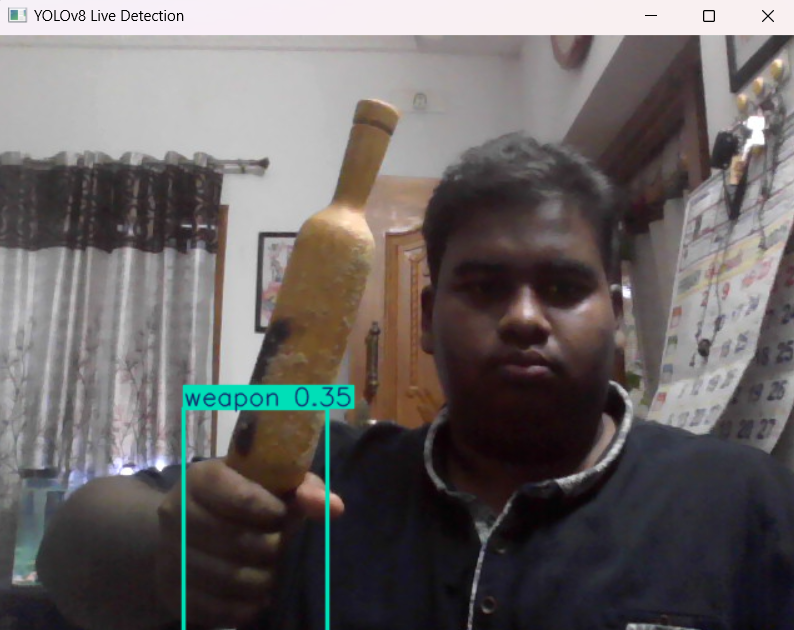
Jeevan Sendur G

Zeex.AI - Coding Round Tasks

Task 1: Training a custom YOLOv8x model using roboflow dataset,

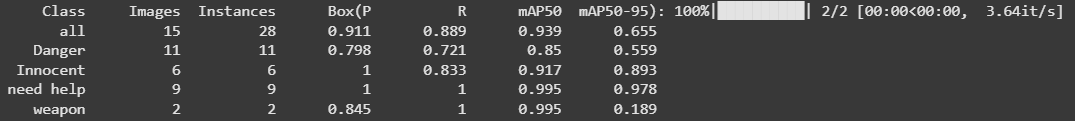
Dataset Link: <https://universe.roboflow.com/jeevan-sendur-g-workspace/act-prediction-kmshx>

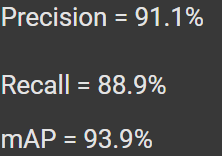
Testing with camera:



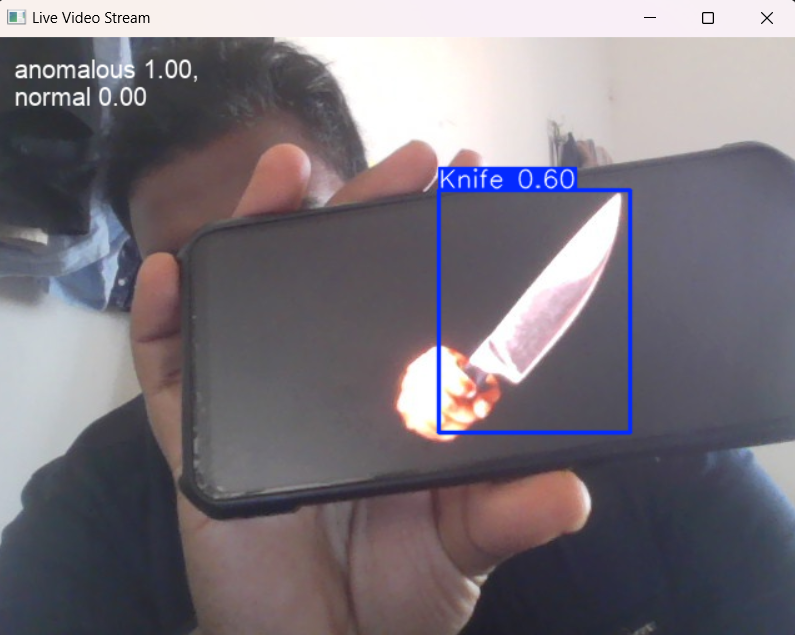
RUN testing.py TO GET INFERENCE, PRES ESC TO EXIT.

Metrics:

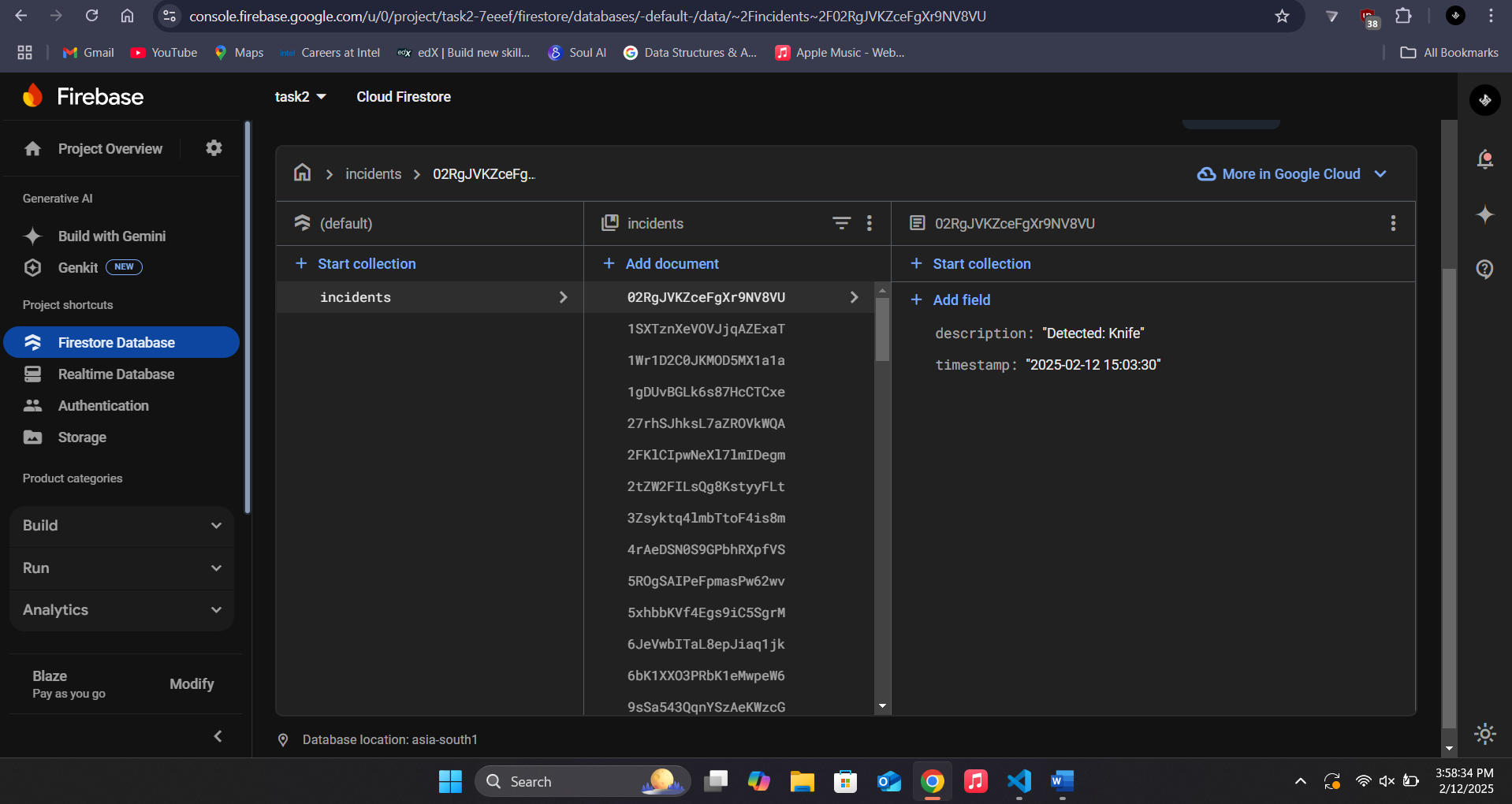




Task 2. Multi-model integration and getting inference over live-streamed footage, and sending alerts to a real-time database.  
  
Dataset Link of anomaly detection model: <https://universe.roboflow.com/jeevan-sendur-g-workspace/anomaly-n6o0c>



Updating Timestamp and detected objects to firestore database



Run client and server in separate terminals, change my IP to yours.

Press ESC to exit, or close terminal.

Task 3, RL via HF to fine-tune yolo model.

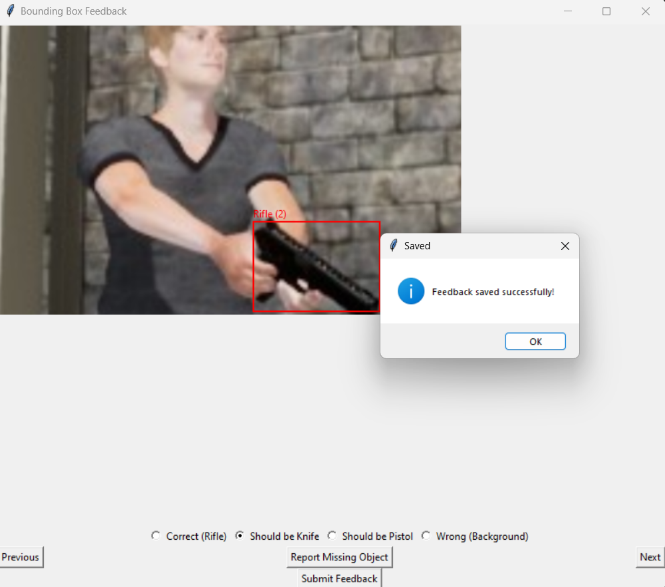
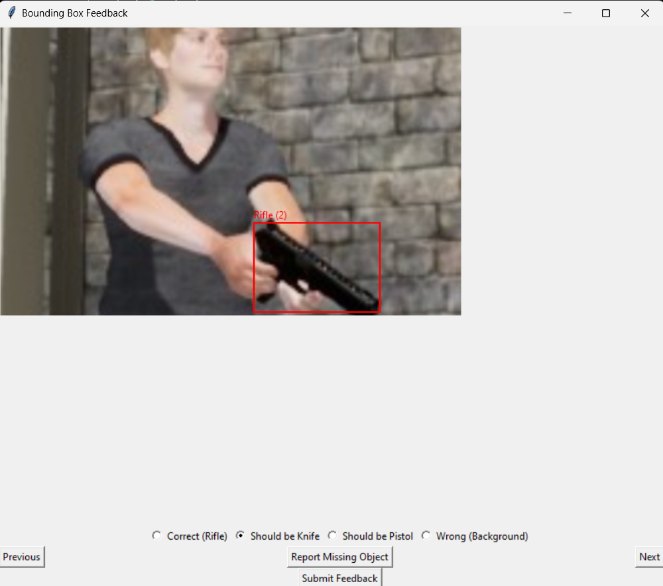
An image or a video is passed, videos are samples at the rate of 2 fps

Human feedback is collected from 4 buttons for each box in each image.  
1. Correct prediction(image is stored to dataset without any mutations)

2. should be <alternative 1>(class is changed, and stored in dataset)

3. should be <alternative 2>(class is changed, and stored in dataset)

4. Background (It is not stored, confidence threshold is increased). Submit feedback and press ‘x’ to quit



Reward signal  
  
images and labels.txt are stored in the dataset, ready to be fine-tuned again  
  
if too many false positives (background as any class) are detected, confidence threshold is increased

Section 2:

Overcoming “Catastropic forgetting” while finetuning

Adding New data to only one class

from the above metrics, we can see that knife is poorly performing

possible fine-tuning stategies

1. Add an equal amount of 'exemplar' older data of other classes, to avoid catastropic forgetting, also called 'replay method'

2. Reduce LR to the range of 0.000001 to 0.00001 to prevent drastic changes in weights to improve knowledge retention

3. gradual unfreezing of backbone layers while fine-tuning, freezing backbone layers first, gradually unfreezing them

Best Approach:

Using low LR(AdamW optimizer)

Freezing/unfreezing conv layers over epochs

I have merged both older dataset and newer dataset, and implemented replay method.

Recommended Methods

Its better to annotate data ourselves, this can be done on roboflow.

train image augmentations are important, such as grayscale, flip, etc.

Use ultralytics API to generate dataset, and train on Colab

Always use latest version of YOLO. currently it is yolo 11

Resources

[https://docs.ultralytics.com/](https://www.google.com/url?q=https%3A%2F%2Fdocs.ultralytics.com%2F)