2024 spring CV final project – report

R12942102 黃顥

R12942009 廖珀毅

M11202222 楊凱程

I. Abstract

In this report, we provide an algorithm that can detect the frames in which the doors of public transport vehicles are in the process of opening or closing. The method we use is accomplished through the YOLOv5 model, and by observing videos from Samples and Tests, we finely tune the judgment of the frames for opening and closing.

II. Method

In this section, we'll give a detailed explanation of proposed method.

2.1 YOLOv5

In this project, we primarily utilized the YOLOv5 model to initially determine on a broad scale whether the door in each frame is open or closed. We first converted the sample videos into individual frame image files using Python. Each image was then annotated with a bounding box and labeled according to the current state of the door (open or closed). These labels were saved in label files, which were subsequently used to train the model.

2.2 Algorithm for determining the frame numbers when the door is. opening or closing.

First, we use `cv2.VideoCapture` to load the video, which allows us to obtain the frame count for subsequent processing. Additionally, since some videos perform better after being rotated, we also incorporate `cv2.rotate` here to achieve higher accuracy.

We load a pre-trained model using `torch.hub.load`, adding our trained `best.pt` file. Each frame image is input into the model, and the output includes the object's bounding box, confidence level, and the category of the bounding box (Open or Close).

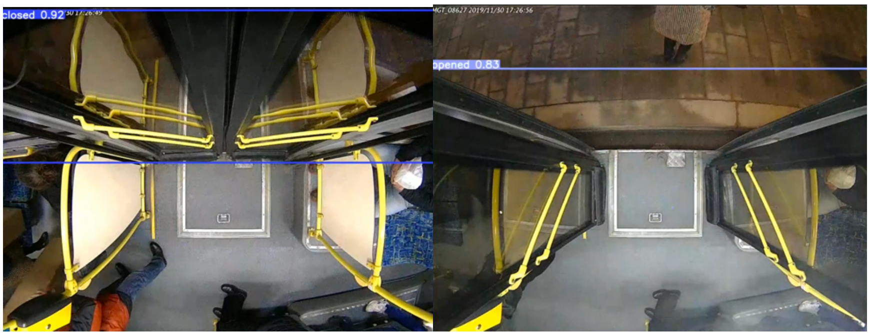
Next, we use the bounding boxes and confidence levels obtained earlier to determine whether each frame represents a door opening or closing. Our model performs better at detecting the state of a door being closed, so we primarily use the confidence level of the door being closed for this determination. By setting a threshold, if the confidence level is above the threshold, it indicates that the door is closed; if it is below the threshold, it suggests the door is not closed, allowing us to infer that the door may be open. Additionally, we have set other thresholds to exclude data with too low confidence levels, helping to smooth the curve.

After determining whether each frame represents a door opening or closing, the next step is to identify the frames for opening and closing actions. We use `np.diff` and `np.where` to perform differencing (comparing with the immediate right neighbor) to pinpoint frames where there is a change in class, treating these as initial peaks. Subsequently, we apply additional criteria to determine if these peaks are genuine, which include:

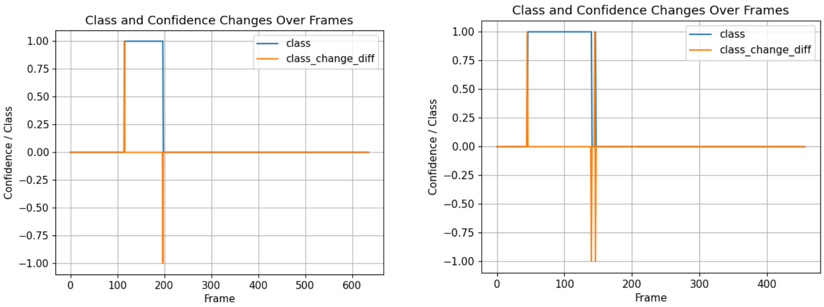
1. Boundary Check: We first check if any peaks are at the beginning or end of the video. If there are not enough frames around a peak to make an informed decision, then that peak is considered invalid.
2. Initial State Assumption: Since we can assume that the door is closed at the start of the video, the first peak we need to find should indicate an opening. By calculating the count of real peaks (initially assumed to be 0) and taking modulo 2, we can determine whether we are looking for an opening or closing peak.
3. State Consistency Around Peaks: For a valid opening peak, all frames to the right should indicate the door is open; conversely, for a closing peak, all frames to the left should indicate the door is open. Based on this, we decide whether to place the observation window on the left or right side of the peak.
4. Window Observation: By observing the states within the window—where open is represented as 1 and closed as 0—we calculate the

average to estimate the proportion of closed states within the window. If there are too many closed states (i.e., the average is too low), then this peak is also not considered genuine.

These steps ensure that only significant changes in door status are recorded as true peaks, enhancing the accuracy of our frame-by-frame analysis of door opening and closing actions. This method is critical in maintaining high precision in our model's performance and delivering reliable results in practical applications.



Output after sending the video to the model



Display the class of each frame and the changes in class.

III. Conclusion

In summary, our algorithm utilizes a trained model to initially predict whether a video frame indicates a door opening or closing. It then employs a series of conditions to sift through and identify the true peaks, ultimately determining the frames for opening and closing. Looking forward, we believe improvements can be made by applying more pre-processing to the videos. For example, we could filter out dark or black areas of the video (as most doors have black parts), then use edge detection to identify edges.

Furthermore, after detecting edges, we could use morphological operations (such as Dilation and Erosion) to enhance the integrity of these edges. The goal is to minimize the presence of irrelevant objects in the frame as much as possible, which would simplify the visual data significantly.