STOPSIGN DETECTION BY USING DASHCAMERA VIDEO

To create a powerful ML model for recognizing stop signs in live dash cam footage using pretrained YOLO (You Only Look Once) model, you can follow these steps:

1. Install necessary libraries:
   Make sure you have Python and the required libraries installed. You can use the following:

   ```bash
   pip install opencv-python-headless
   pip install numpy
   ```

2. Download the YOLOv3 weights and configuration files:
   You'll need the YOLOv3 weights (`yolov3.weights`) and configuration file (`yolov3.cfg`). You can download them from the official YOLO website or from a repository that hosts YOLOv3 weights.

3. Create a Python script for object detection:
   You can create a Python script to load the YOLOv3 model, process the dash cam footage, and detect stop signs. Here's a basic example:

   ```python
   import cv2
   import numpy as np

   # Load YOLO
   net = cv2.dnn.readNet('yolov3.weights', 'yolov3.cfg')

   # Load COCO class labels
   classes = []
   with open('coco.names', 'r') as f:
       classes = f.read().strip().split('\n')

   # Load dash cam footage (replace with your video source)
   cap = cv2.VideoCapture('dash_cam_video.mp4')

   while True:
       ret, frame = cap.read()
       if not ret:
           break
   ```

```
# Detecting objects in the frame
blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
net.setInput(blob)
outs = net.forward(net.getUnconnectedOutLayersNames())

# Processing detected objects
class_ids = []
confidences = []
boxes = []
for out in outs:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > 0.5:  # You can adjust this threshold
            # Object detected
            center_x = int(detection[0] * frame.shape[1])
            center_y = int(detection[1] * frame.shape[0])
            w = int(detection[2] * frame.shape[1])
            h = int(detection[3] * frame.shape[0])

            # Rectangle coordinates
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)

            boxes.append([x, y, w, h])
            confidences.append(float(confidence))
            class_ids.append(class_id)

# Non-maximum suppression to eliminate duplicate detections
indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

for i in range(len(boxes)):
    if i in indexes:
        x, y, w, h = boxes[i]
        label = str(classes[class_ids[i]])
        if label == 'stop sign':
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
            cv2.putText(frame, label, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

# Display the frame with detections
cv2.imshow('Dash Cam', frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
```

```
   # Release the video capture and close the window
   cap.release()
   cv2.destroyAllWindows()
```

Make sure to replace `'yolov3.weights'`, `'yolov3.cfg'`, `'coco.names'`, and `'dash_cam_video.mp4'` with the appropriate file paths.

4. Run the script:
   Save the script and run it using Python. The script will process the dash cam footage, detect stop signs, and display the video with bounding boxes around the detected signs.

Note: You may need to fine-tune the YOLO model for better accuracy on stop sign detection, depending on the quality and conditions of your dash cam footage. Additionally, consider integrating this with a warning system for real-world applications.