```
import cv2
import numpy as np

cap = cv2.VideoCapture(0)

if not cap.isOpened():
    print("Camera not opened")
    exit()

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

    h, w = frame.shape[:2]

    # 1. ROI (only the bottom half)
    roi = frame[h//2:h, :]

    # 2. HSV for blue detection
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    lower_blue = np.array([90, 60, 60])
    upper_blue = np.array([130, 255, 255])
    mask = cv2.inRange(hsv, lower_blue, upper_blue)

    # 3. edge
    edges = cv2.Canny(mask, 50, 150)

    # 4. HoughLinesP
    lines = cv2.HoughLinesP(
        edges,
        rho=1,
        theta=np.pi / 180,
        threshold=40,
        minLineLength=30,
        maxLineGap=20
    )

    left_lines = []
    right_lines = []
    cx = w // 2

    if lines is not None:
        for l in lines:
            x1, y1, x2, y2 = l[0]
            mid_x = (x1 + x2) // 2

            if mid_x < cx:
                left_lines.append((x1, y1, x2, y2))
            else:
                right_lines.append((x1, y1, x2, y2))

    # 5. Right and left
    for x1, y1, x2, y2 in left_lines:
        cv2.line(roi, (x1, y1), (x2, y2), (255, 0, 0), 2)

    for x1, y1, x2, y2 in right_lines:
        cv2.line(roi, (x1, y1), (x2, y2), (0, 0, 255), 2)

    # 6. Center line
    if left_lines and right_lines:
        y = roi.shape[0] - 20

        left_x = np.mean([x1 for x1,_,_,_ in left_lines])
        right_x = np.mean([x1 for x1,_,_,_ in right_lines])
        center_x = int((left_x + right_x) / 2)

        cv2.line(roi, (center_x, 0), (center_x, roi.shape[0]), (0, 255, 0), 3)

    # back to the frame
    frame[h//2:h, :] = roi

    cv2.imshow("Parallel Line Detection", frame)

    if cv2.waitKey(1):
        break

cap.release()
cv2.destroyAllWindows()
```