```
while True:
    # Looping the video
    if cap.get(cv2.CAP_PROP_POS_FRAMES) == cap.get(cv2.CAP_PROP_FRAM
        cap.set(cv2.CAP_PROP_POS_FRAMES, 0)
    # Reading frame by frame from video
    success, img = cap.read()
    # Converting to gray scale image
    imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    imgBlur = cv2.GaussianBlur(imgGray, (3, 3), 1)  # Applying blur
    # Applying threshold to the image
    imgThreshold = cv2.adaptiveThreshold(imgBlur, 255, cv2.ADAPTIVE_
                                         cv2.THRESH_BINARY_INV, 25,
    imgMedian = cv2.medianBlur(imgThreshold, 5)  # Applying blur to
    kernel = np.ones((3, 3), np.uint8)
    imgDilate = cv2.dilate(imgMedian, kernel, iterations=1)
    # Passing dilate image to the function
    checkParkingSpace(imgDilate)
    cv2.imshow("Image", img)
    cv2.waitKey(10)
```