MANUAL CODE REVIEW ERRORS FOUND IN THE WEB APP CODE:

Face-Counter

• No input check for `img` in `findFaceMesh`.
• `cv2.VideoCapture(0)` does not have error handling.
• It does not check whether a frame has been captured correctly as indicated by the parameter `success`.
• Access `multi_face_landmarks` without checking if `self.results` is `None`.
• It does not check whether the list `faces` may be empty before it tries to get the first element, `faces[0]`.
• Parameters of `cv2.putText` hardcoded without providing a way to configure it from outside.
• The main loop lacks cleanup or proper exception handling.
• Lack of logging or error reporting mechanism.
• It doesn't take thread-safety into consideration regarding shared resources, such as `cap`.
• There are magic numbers used, such as thickness and circle_radius, without explanation.

Virtual Mouse
• No validation of the input `frame` after reading from `cap`.
• No exception handling while calling the function `pyautogui.press()`.
• No checking for `None` value of `results` before working with `multi_hand_landmarks`.
• Misusing `pyautogui` for control by changing volume without the confirmation of users.
• The parameters are hardcoded for initializing `Hands` without explanation.
• No cleanup or exception handling in the main loop.
• The code is repeated since the entire script is duplicated.
 No acton logging is done. Examples include volume changes.
 No gesture threshold or sensitivity setting.

- `img` input validation not done in the `findHands` method.
- Error handling of `cap.read()`.
- Check if `self.results` is not None before trying to access `multi_hand_landmarks`.
- Use `autopy.mouse.move` without boundary checks, which might lead to misuses.
- Exception handling when getting screen dimensions by `autopy.screen.size()`.

X 

SEAN MAVHIMA

Intern