```python
import cv2
from cvzone.HandTrackingModule import HandDetector

# Open the camera
cap = cv2.VideoCapture(0)
cap.set(3, 1280)
cap.set(4, 720)

# Initialize Hand Detector
detector = HandDetector(detectionCon=0.7)

while True:
    success, img = cap.read()
    hands, img = detector.findHands(img)

    # Read an image for overlay and resize it to match the video feed
    img1 = cv2.imread(r"C:\Users\rkssp\Desktop\bharathi.jpg")
    img1 = cv2.resize(img1, (img.shape[1], img.shape[0]))  # Match the video feed dimensions

    if len(hands) == 2:
        hand1 = hands[0]
        hand2 = hands[1]

        # Calculate the distance between the hands
        length, _, _ = detector.findDistance(hand1["center"], hand2["center"], img)

        # Adjust the scaling factor based on the distance
        scale = int((length - 50) // 2)  # Adjust the divisor as needed for a smoother zoom

        # Resize the overlay image while preserving color
        h1, w1, _ = img1.shape
        newH, newW = h1 + scale, w1 + scale
        img1_resized = cv2.resize(img1, (newW, newH))

        # Define the region to replace with the resized image
        roi = img[0: img.shape[0], 0: img.shape[1]]

        # Ensure both images have the same channels
        img1_resized = img1_resized[:roi.shape[0], :roi.shape[1]]

        # Blend the resized image with the original image using the mask
        img[0: img.shape[0], 0: img.shape[1]] = img1_resized

    # Display the result
    cv2.imshow("Hand Tracking and Zooming", img)

    # Break the loop if 'q' is pressed
```

```
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release resources
cap.release()
cv2.destroyAllWindows()


```