

```
1 import cv2
2 from cvzone.HandTrackingModule import HandDetector
3
4 # Open the camera
5 cap = cv2.VideoCapture(0)
6 cap.set(3, 1280)
7 cap.set(4, 720)
8
9 # Initialize Hand Detector
10 detector = HandDetector(detectionCon=0.7)
11
12 while True:
13     success, img = cap.read()
14     hands, img = detector.findHands(img)
15
16     # Read an image for overlay and resize it to match the video feed
17     img1 = cv2.imread(r"C:\Users\rkssp\Desktop\bharathi.jpg")
18     img1 = cv2.resize(img1, (img.shape[1], img.shape[0])) # Match the video feed dimensions
19
20     if len(hands) == 2:
21         hand1 = hands[0]
22         hand2 = hands[1]
23
24         # Calculate the distance between the hands
25         length, _, _ = detector.findDistance(hand1["center"], hand2["center"], img)
26
27         # Adjust the scaling factor based on the distance
28         scale = int((length - 50) // 2) # Adjust the divisor as needed for a smoother zoom
29
30         # Resize the overlay image while preserving color
31         h1, w1, _ = img1.shape
32         newH, newW = h1 + scale, w1 + scale
33         img1_resized = cv2.resize(img1, (newW, newH))
34
35         # Ensure both images have the same channels
36         img1_resized = img1_resized[:img.shape[0], :img.shape[1]]
37
38         # Blend the resized image with the original image using the mask
39         img = cv2.addWeighted(img, 1, img1_resized, 0.5, 0)
40
41         # Display the result
42         cv2.imshow("Hand Tracking and Zooming", img)
43
44         # Break the loop if 'q' is pressed
45         if cv2.waitKey(1) & 0xFF == ord('q'):
46             break
47
```

```
48 # Release resources
49 cap.release()
50 cv2.destroyAllWindows()
51
```