



# Hand Gesture Recognition Using Computer Vision

**Presented by**

Midhun P (TVE17EC027)

Sreejith S (TVE17EC049)

Vipin Chandran M (TVE17EC061)

# Our Project so far

- ✓ In the first phase of our project, we were able to detect a hand (the colour of skin) from an image.



Phase 1 results

- ✓ Adding onto that, in the second phase, we recognized different hand gestures from a live video.



Phase 2 results

# Phase 3

## Objective –

The intention of phase 3 of our project is to build a virtual mouse which works based on hand movements.

**Algorithm** – (only additions to phase 2 of the project are mentioned below)

- ✓ Pynput library allows us to control input devices like the mouse.
- ✓ The NumPy range for the red color is defined.
- ✓ Convert BGR to HSV, create the mask and perform morphological operations.
- ✓ Draw and count the number of contours.
- ✓ For 2 objects : Release the operation of mouse.
  - > Make a rectangle around the objects and draw a line between them.
  - > Draw a circle at the center of the line that will act as a pointer.
  - > Update mouse location while moving.
- ✓ For 1 object: Perform left click action.
- ✓ Display camera and show required images.

```
In [ ]: import cv2
import numpy as np
from pynput.mouse import Button, Controller
import wx

def leftClick():
    mouse = Controller()
    app = wx.App(False)
    (sx,sy) = wx.GetDisplaySize()
    (camx,camy) = (480,320)

    lower_bound = np.array([130, 90, 109])
    upper_bound = np.array([203, 225, 180])
    cam = cv2.VideoCapture(0)
    cam.set(3, camx)
    cam.set(4, camy)

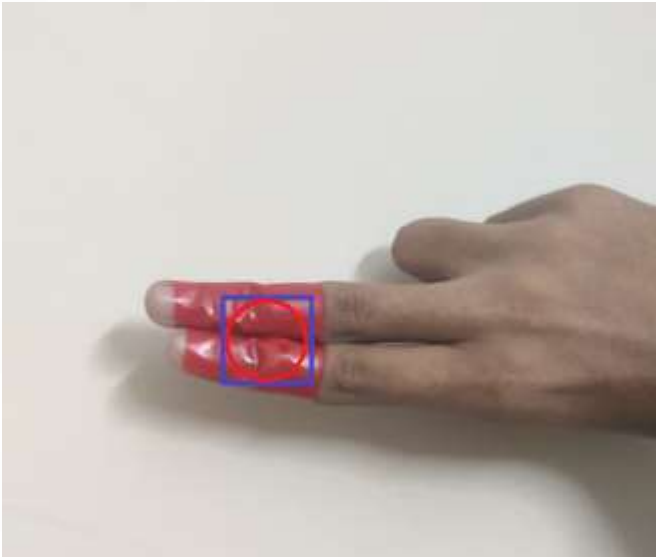
    kernelOpen = np.ones((5,5))
    kernelClose = np.ones((20,20))
    mLocOld = np.array([0,0])
    mouseLoc = np.array([0,0])

    Df = 2.3
    pinchFlag = 0

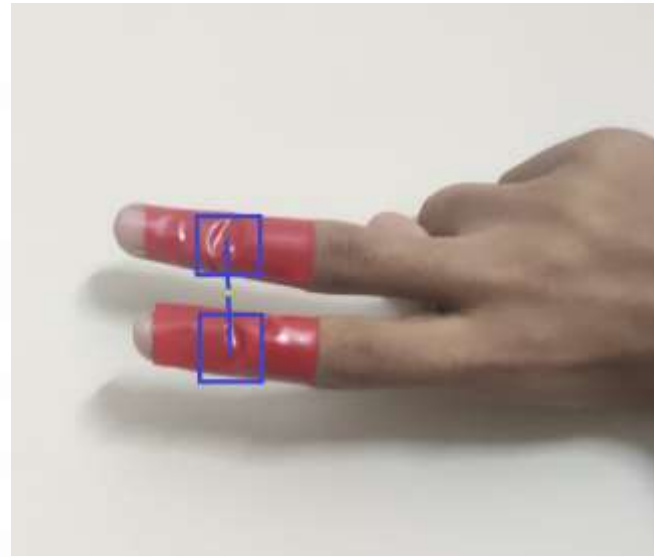
    while True:
        ret,img = cam.read()
        imgHSV = cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
        mask = cv2.inRange(imgHSV,lower_bound,upper_bound)
        maskOpen = cv2.morphologyEx(mask,cv2.MORPH_OPEN,kernelOpen)
        maskClose = cv2.morphologyEx(maskOpen,cv2.MORPH_CLOSE,kernelClose)
        maskFinal = maskClose
        conts,h = cv2.findContours(maskFinal.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)

        if len(conts)==2:
            if pinchFlag == 1:
                mouseLoc = mouseLoc + (conts[0][0][0] - conts[1][0][0]) * Df
            else:
                mouseLoc = conts[0][0][0]
            mouse.click(Button.LEFT, mouseLoc[0], mouseLoc[1])
            pinchFlag = 1
```

## Phase 3 results



Mouse click operation



Mouse pointer

The implementation of our virtual mouse is illustrated in the video file attached.





Thank  
you!

The image features a central teal circle with a rough, hand-painted brushstroke texture. Inside this circle, the words "Thank you!" are written in a white, elegant cursive script. The background is white, with decorative clusters of blue squares in the top-left and bottom-right corners. These squares vary in size and shade, ranging from light sky blue to deep navy blue, creating a modern, geometric pattern.