```python
import mediapipe as mp
import cv2
import threading
import socket
import time


host = '0.0.0.0'
port = 8889
#host = '192.168.10.2'
#port = 8889
locaddr = (host, port)



# Create a UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

tello_address = ('192.168.10.1', 8889)

sock.bind(locaddr)

def recv():
    while True:
        try:
            data, server = sock.recvfrom(1518)
            print(data.decode(encoding="utf-8"))
        except Exception:
            print('\nExit . . . RECV\n')
            break

recvThread = threading.Thread(target=recv)
recvThread.start()
#send sdk
sock.sendto(b'command', tello_address)
print('command ok')
time.sleep(0.5)

time.sleep(1)
#sock.sendto(b'takeoff', tello_address)
#time.sleep(2)
#sock.sendto(b'up 80', tello_address)
#time.sleep(2)
sock.sendto(b'streamon', tello_address)
print('stream on')

LOCAL_IP = '192.168.10.1'
LOCAL_PORT ='11111'
addr = ("udp://%s:%s?overrun_nonfatal=1&fifo_size=50000000" % ('192.168.10.1',
'11111'))
cap = cv2.VideoCapture(addr)
cap.set(cv2.CAP_PROP_BUFFERSIZE, 2)

print('start cap')

mpHands = mp.solutions.hands
hands = mpHands.Hands()
mpDraw = mp.solutions.drawing_utils
while True:
```

```
    ret, frame = cap.read()
#   frame = cv2.flip(frame, 1)     #映像を左右反転させる。
    if frame is None or frame.size == 0:
            continue
    resize_frame = cv2.resize(frame, dsize=(480, 360))
    imageRGB = cv2.cvtColor(resize_frame, cv2.COLOR_BGR2RGB)

    results = hands.process(imageRGB)
    if results.multi_hand_landmarks:
        for handLms in results.multi_hand_landmarks:
            for id, lm in enumerate(handLms.landmark):
                h, w, c = frame.shape
                cx, cy = int(lm.x * w), int(lm.y * h)
            mpDraw.draw_landmarks(resize_frame, handLms, mpHands.HAND_CONNECTIONS)

    lmList = []
    handNo = 0
    draw = True
    if results.multi_hand_landmarks:
        myHand = results.multi_hand_landmarks[handNo]
        for handLms in results.multi_hand_landmarks:            #landmark
            for id, lm in enumerate(handLms.landmark):
                #print(id, lm)
                h, w, c = frame.shape
                cx, cy = int(lm.x * w), int(lm.y * h)
                #print(id, cx, cy)
                lmList.append([id, cx, cy])
                #print(lmList)

    #tipIds = [4, 8, 12, 16, 20]

    if len(lmList) != 0:
        if lmList[8][2] < lmList[6][2]:    #人差し指
            sock.sendto(b'takeoff', tello_address)
            print('takeoff')

        #if lmList[4][2] < lmList[2][2]:      #親指  認識が難しかったです。
        #    print('down')
        if lmList[12][2] < lmList[10][2]:  #中指
            sock.sendto(b'forward 30', tello_address)
            print('forward')
        #    time.sleep(2)
        #if lmList[16][2] < lmList[14][2]:      #薬指  関節が硬いので立たせるのが困難
         #   print('left')
        if lmList[20][2] < lmList[18][2]:  #小指
            sock.sendto(b'back 30', tello_address)
            print('back')



    cv2.imshow("Output", resize_frame)
    if cv2.waitKey(1) & 0xff == 27:
        break
sock.sendto(b'land', tello_address)
cap.release()
cv2.destroyAllWindows()
sock.close()
```