

Hand Pose Detection System + Gesture Control

Project Report



***TITLE PAGE OF DIP REPORT***

|  |  |
| --- | --- |
| **Course Number and Name:**  Digital Image Processing | |
| **Semester and Year:**  5th semester Fall 2021 | |
| **Name of Student:**  Hameed Ur Rehman & Ahmad Hasham  Fa19-BCS-100 & Fa19-BCS-028 | **Name of Instructor:**  Doctor Muhammad Sharif |
| **Date of Report Submitted:**  16th-December-2021 | **Grade:** |
|  |  |

**We divided our Project into five different tasks:**

|  |
| --- |
| 1. **DEADLINES** |
| **Task 1: 14 October**  Proposal of our Project/designing algorithm  **Task 2: 28 October**  Basic Hand Tracking System/loading image and getting predictions  **Task 3: 11 November**  Recognize different fingers count and other poses/show detections  **Task 4: 25 November**  Perform tasks on computer without touching like increase and decrease of brightness and volume/drawing skeleton  **Task 5: 16 December**  Control our page upside down in our computer/complete project |

**Summary:**

A robust Hand Pose Detection system will be developed along with functionality like controlling options on our computer. The proposed method uses an algorithm (open pose) which detects key points in a hand. Hand Key point detection is the process of finding the joints on the fingers as well as the fingertips in each image. The algorithm takes an image or a video as input and provides the result with the key points highlighted. The output will be having a skeleton like structure outlined from the hand.

**Introduction:**

In real world situation, hand gestures are important in a noisy area where other person can’t hear or any person who is deaf or swallowing impaired has a need of sign language, so they communicate through hand gestures. In computers, hand gesture recognition was invented as early as 1980s. Users can use simple gestures to control or interact with devices without physically touching them. On the large scale, there are gesture recognition software and libraries and algorithms of different hand gestures, and we implement them through code. By the way, there are two types of algorithms 3D model-based algorithms and skeleton-based algorithms. Skeleton-based are much easy to use. These impaired people need gestures to understand each other so, in some situations we need hand tracking to help computer understand hand gestures. Also, we need hand gestures for the augmented/virtual reality which is a modern innovation. We can play different games in Play Station 5 through hand gesture recognition. The technology that is used is motion sensor in the cameras. Depth data is now used in recent times for more improved gestures tracking. In my opinion, hand gestures may change the way we communicate through the computer. In the long run hand gestures may completely replace touch typing as buttons typing was removed by touch typing. It’s a much easier way of communication, we won’t be needing a remote for our TV. We shall use our phones and tablets from some distance. It is not an imagination anymore, a company named Leap Motion last year introduced the Leap Motion Controller, a gesture-based computer interaction system for PC and Mac.

* **Task 1**

In our first most task, we made our **Project Proposal** in which we have given our model of our project that we have 21 finger points and through that we will recognize our hand to the computer. We explained that how our project will detect our different hand poses through libraries media pipe and opencv2. Also, we gave a brief introduction about our project and discussed the future of our project that a robust Hand Pose Detection system will be developed along with functionality like controlling options on our computer. And the figure shown below was our theme.

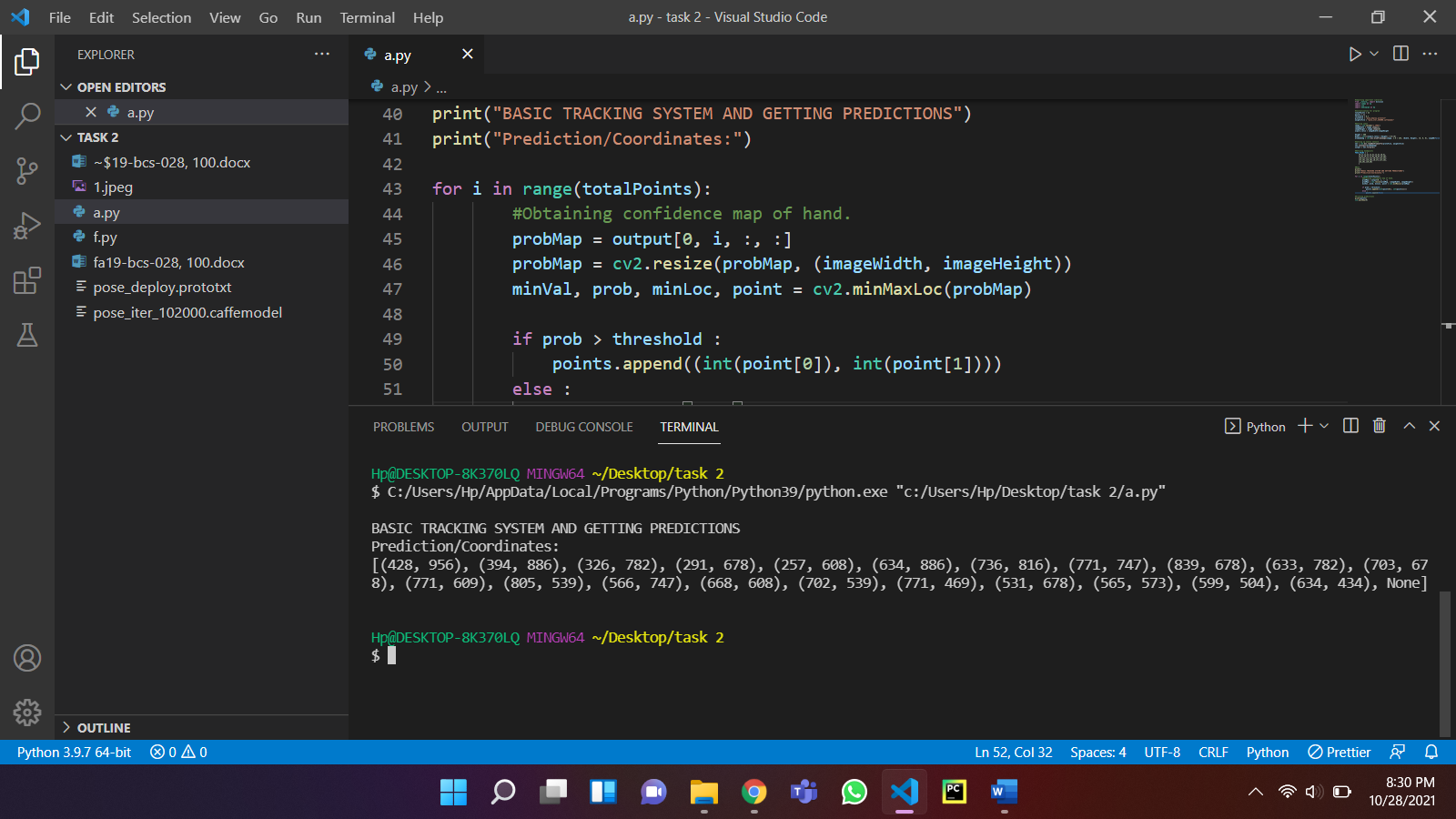
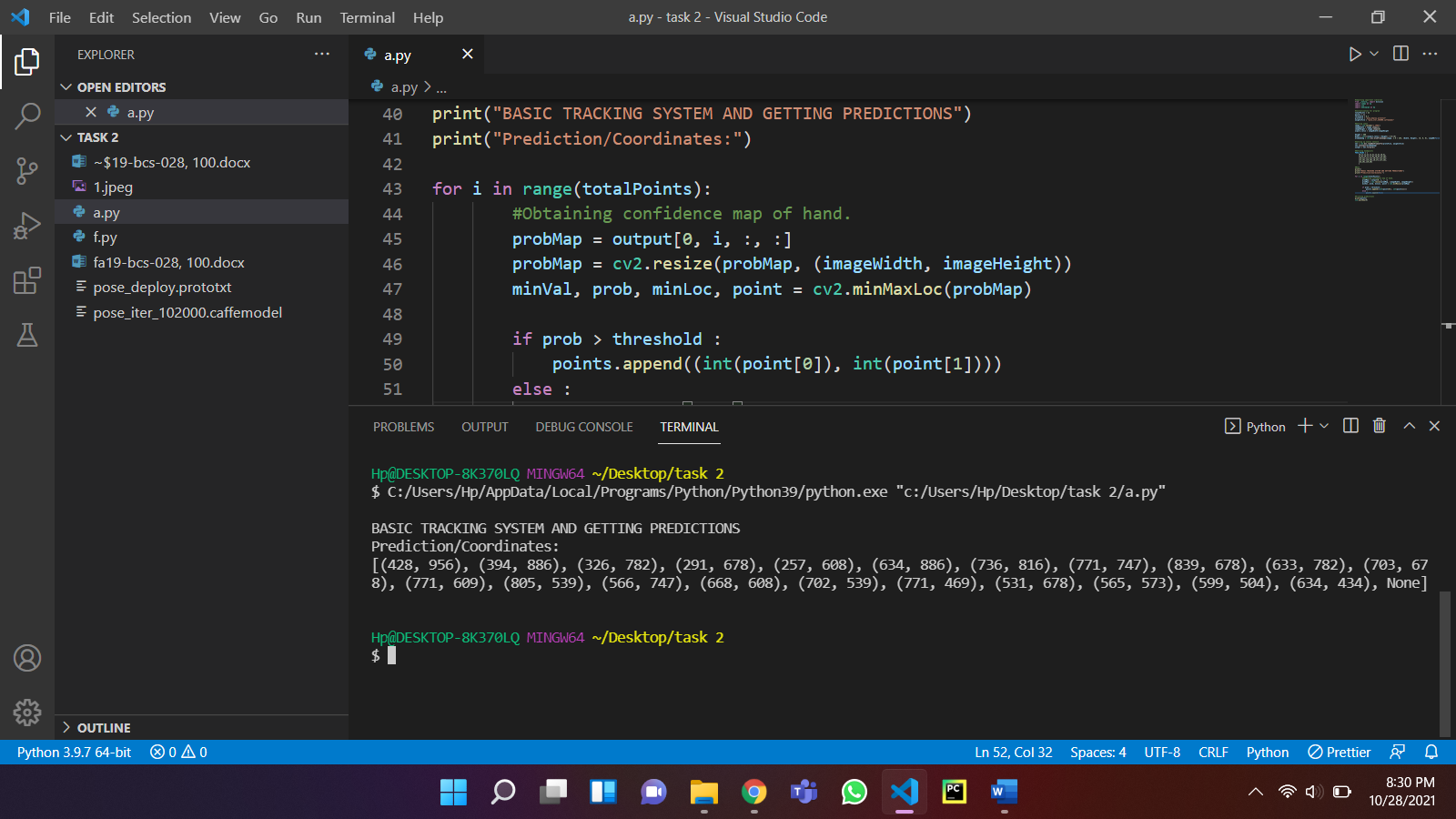
A close-up of a hand

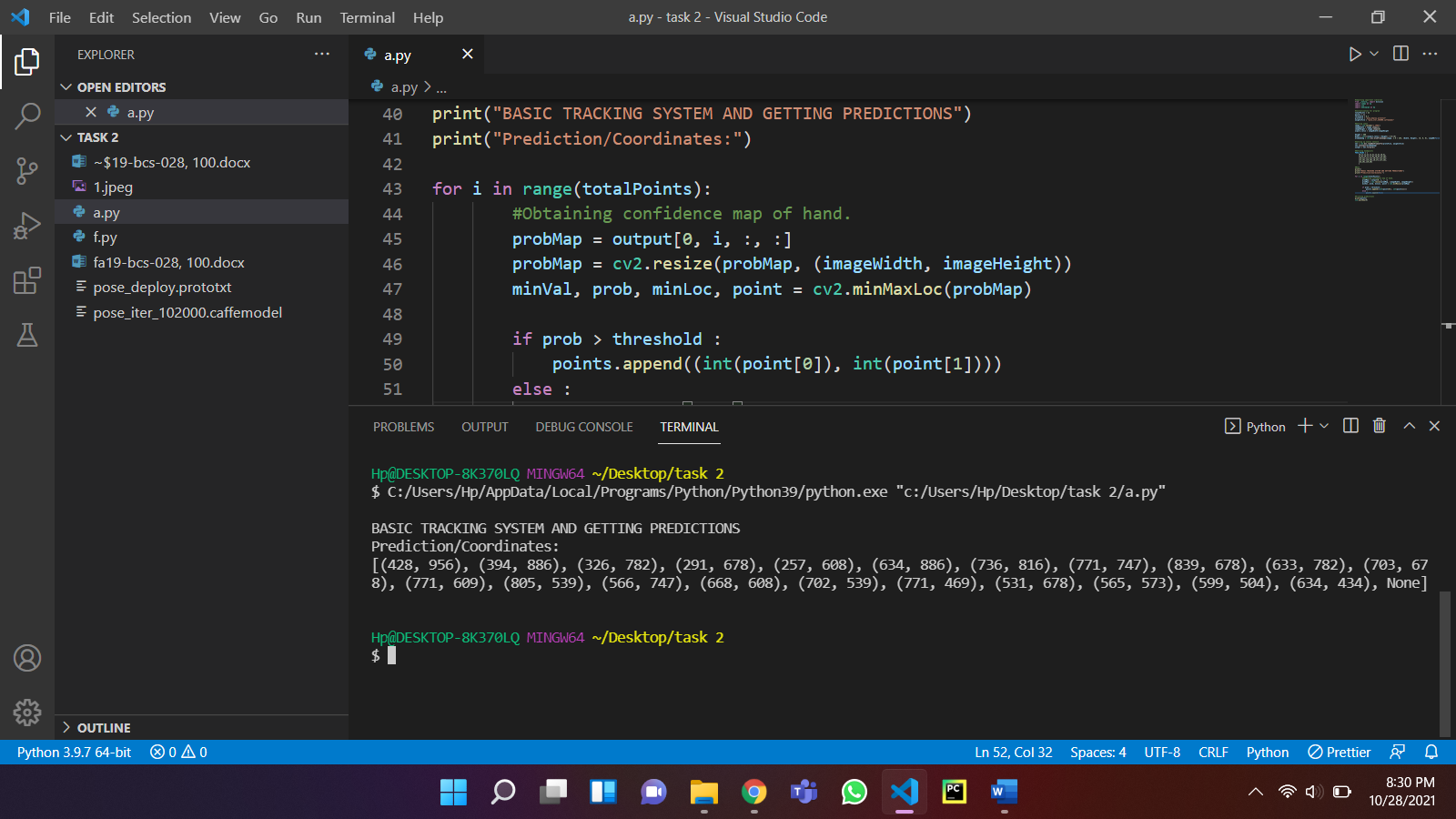
Description automatically generated with low confidence

* **Task 2**

In task 2, we made a **basic hand tracking system.** We designed a basic tracking system for our project Hand Pose Detection and Gesture Control. An image in input in our algorithm which after processing gives us desired output. Our task 2 is obtaining predictions. Here predictions mean the coordinates of the entered image in the frame. A hand has 22 landmarks which is already trained in a model which we imported in this program. We have also scale this program so it can process video inputs and even live inputs from computer’s webcam in the future. The following output represents co ordinates values as we move our hands, values change.

**OUTPUT:**



* **Task 3**

In Task 3, we made computer recognize different **fingers counts and pose detection**. First, OpenCV library is imported. It is necessary for image processing and computer vision. Some other libraries like numpy and mediapipe are also included which are necessary for array manipulation and drawing/sketching on key points of hand respectively. Then some important data is being initialized. We made an empty array named points for storing the predicted points. We looped through the total hand key points which are 22 as in our imported model and performed all the processing. First we were printing the predictions only on the console but now we have a kind of gui interface for our webcam code and we are also saving the outputs of the input image as an image file. The webcam input is take from open cv method of captureVideo. It takes the port number of our webcam as input. It’s usually 0 or 1. We tried both but only 0 worked. Method of mediapipe library are also used like solutions. The solutions method further have functions for pose detections like it can detect head, hand, legs, face, body etc. We accessed it’s hands method and invoked the hands() function. Mediapipe also provides us the ability to draw and sketch. So it becomes easy for us to detect and show detection using it’s drawing utilities. The hands() function takes out webcam input video from video capture and returns the results. We just loop over these obtained results to mark the detections in the hand.

**OUTPUT:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

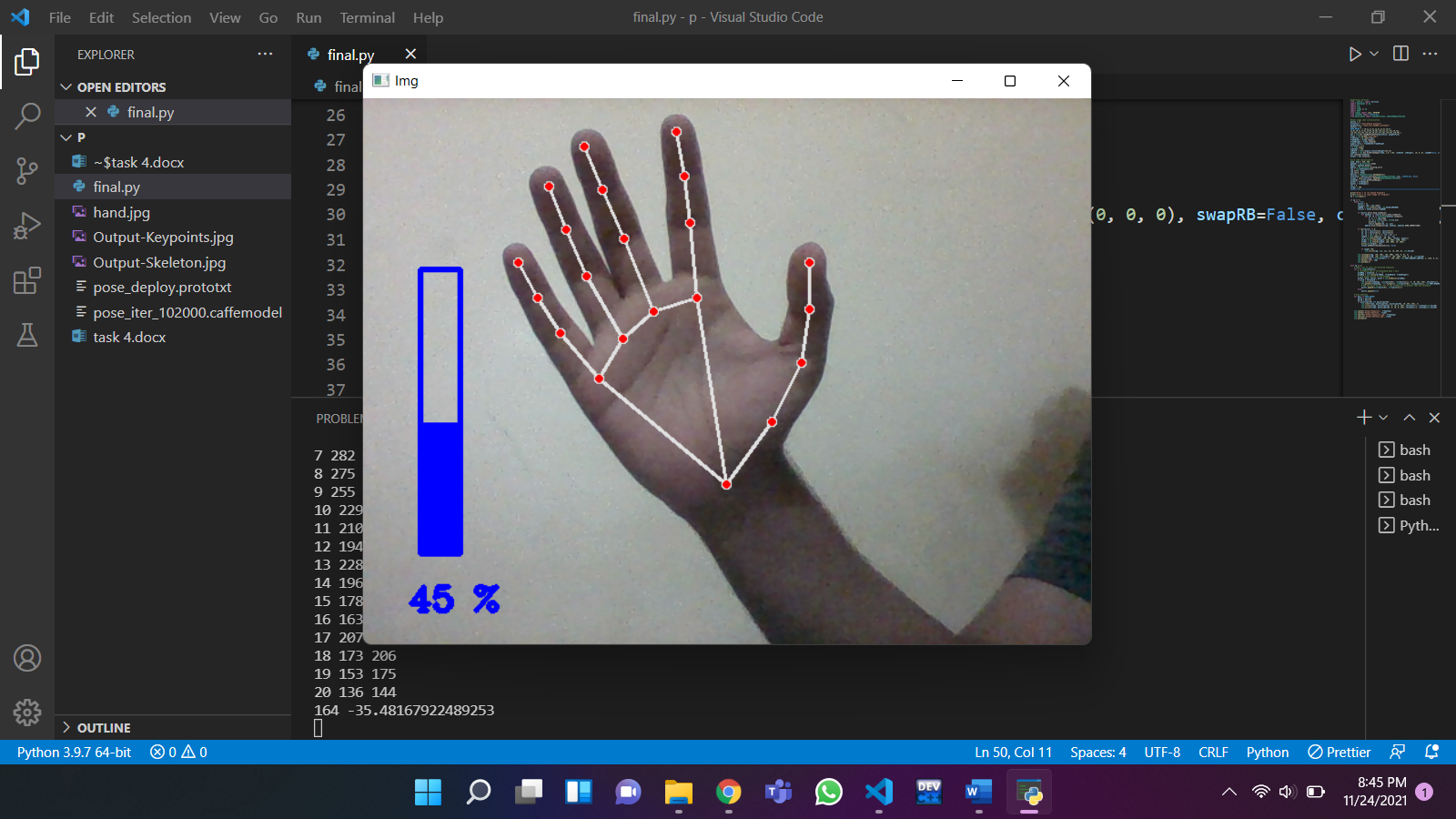
* **Task 4**

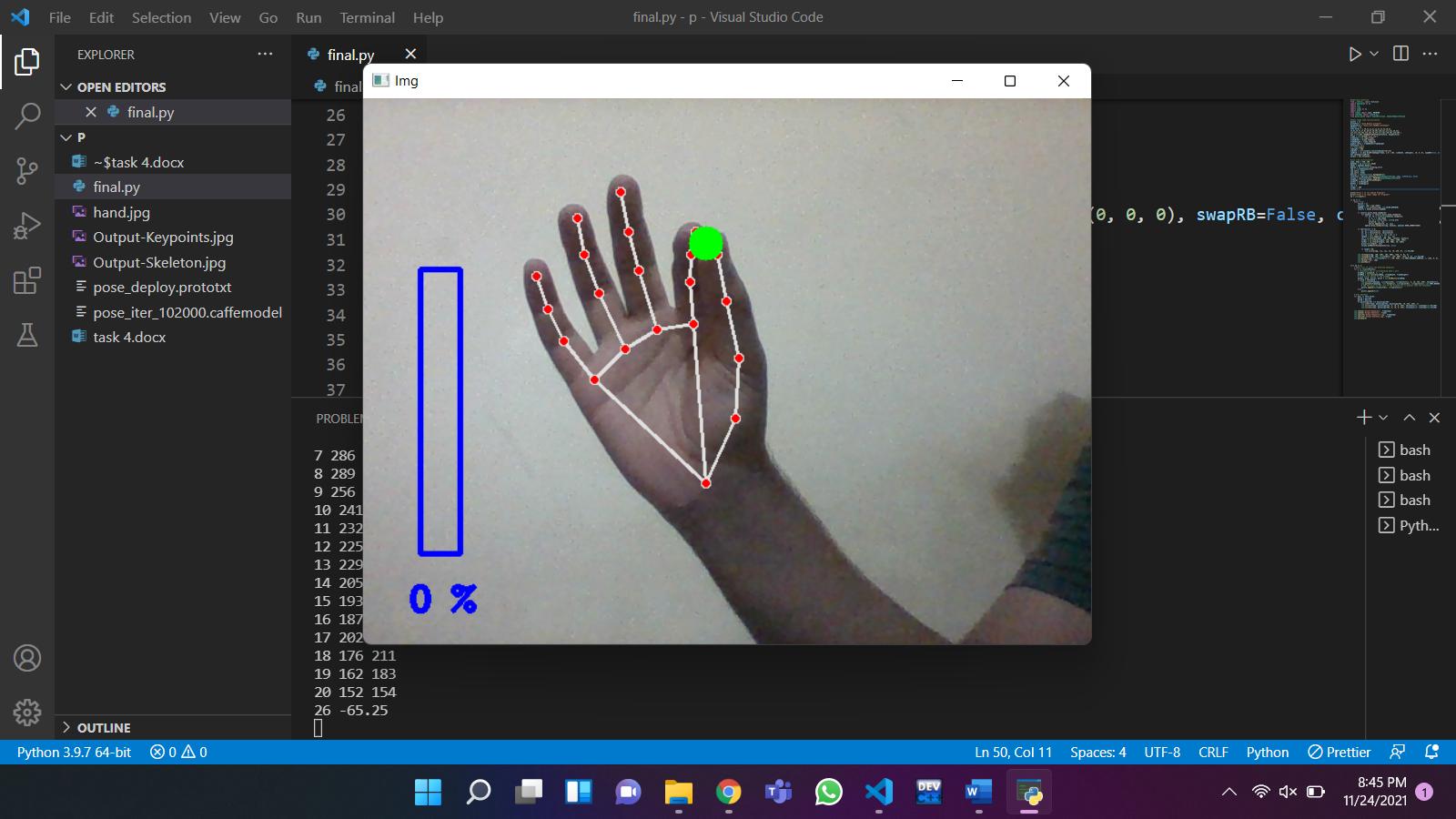
In cmd we installed these libraries:

* + pip install opencv-python
  + pip install mediapipe
  + pip install numpy
  + pip install pycraw
  + pip install ctypes
  + pip install comtypes
  + pip install screen\_brightness\_control
  + Also check for any package that's need to be installed, on cmd
  + then run the scripts python VolumeControl.py

Now, we used volume control library and get the volume range from volume.GetVolumeRange() method. Then we have setted up our webcam as usual. Then using hand landmarks model for identifying hands. Then we have used multi\_hand\_landmarks method for Finding postion of Hand landmarks. We assigned variables for thumb and index finger. Now we have to convert length range into volume range using numpy. Changing system volume and drawing volume bar using their methods. Then displayed output using imshow.

**OUTPUT:**

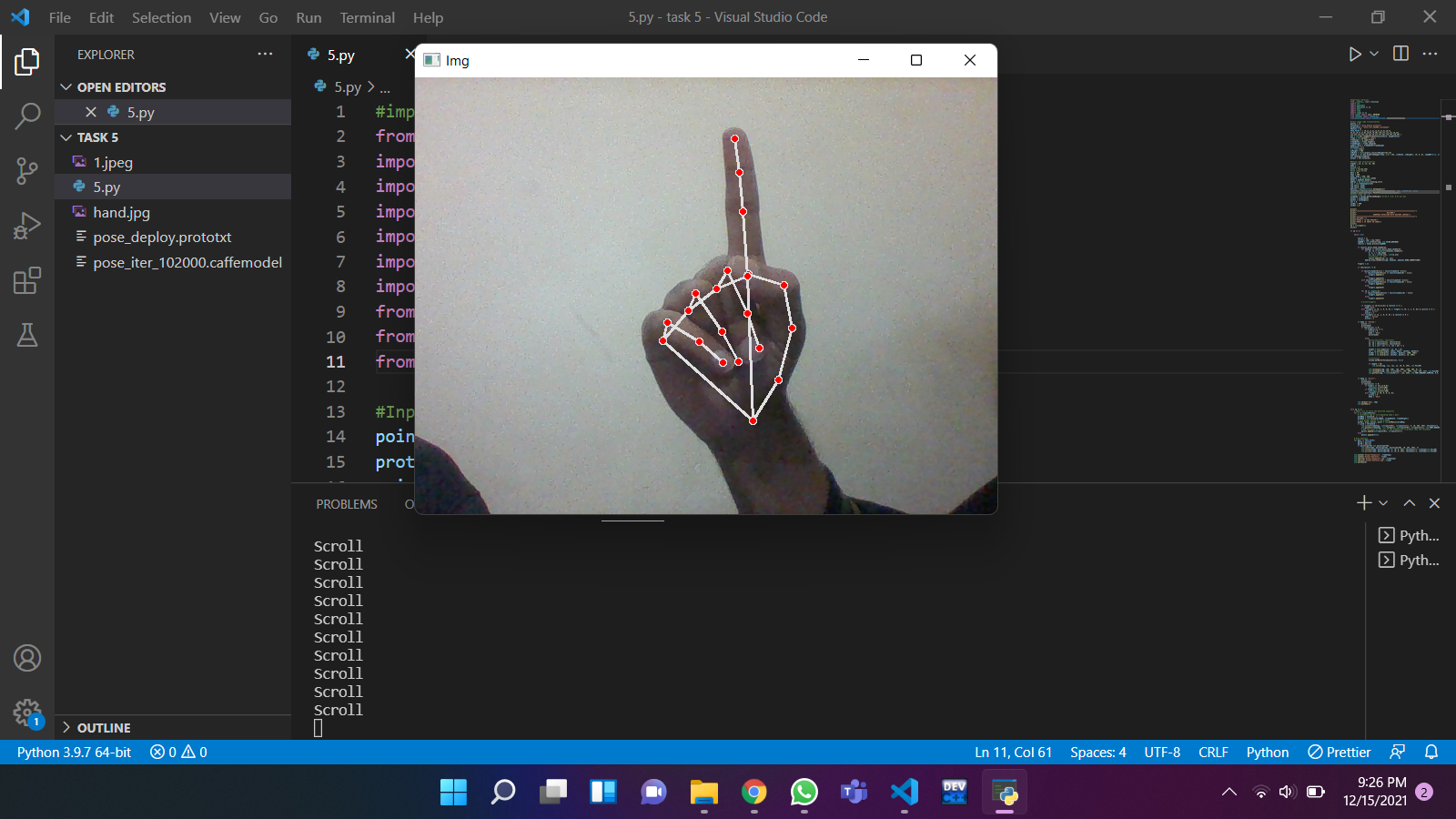


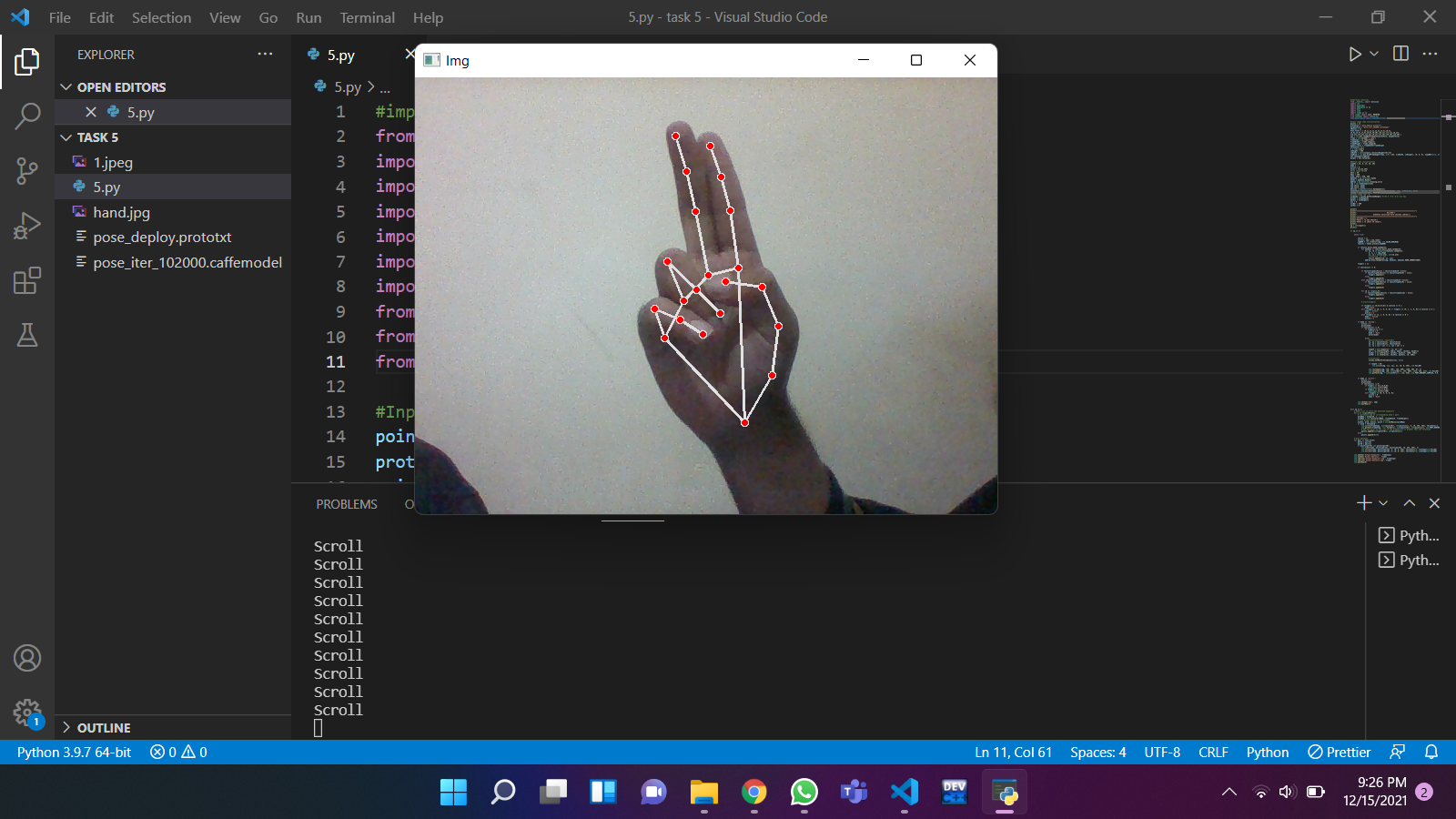


* **Task 5**

In task 5, we **Controlled our page up and down in our computer.** As this was a difficult one, we installed some more libraries for it. Like we used pyautogui library ctypes cast and POINTER libraries. In our final code, we added our scrolling portion with hand detection so we can move our page upside down.

**OUTPUT:**

****

****

**REFERENCES:**

1. <https://github.com/CMU-Perceptual-Computing-Lab/openpose>
2. <https://github.com/spmallick/learnopencv>
3. <https://arxiv.org/pdf/1704.07809.pdf>
4. <https://youtu.be/tVw4SjzBZsU>
5. <https://learnopencv.com/wp-content/uploads/2018/05/hand-output-Skeleton.jpg> <https://www.pexels.com/photo/white-laptop-female-hand-note-pen-phone-desk-6471/>