# CSE-666- BIOMETRICS IMAGE ANALYSIS – ASSIGNMENT-2-REPORT

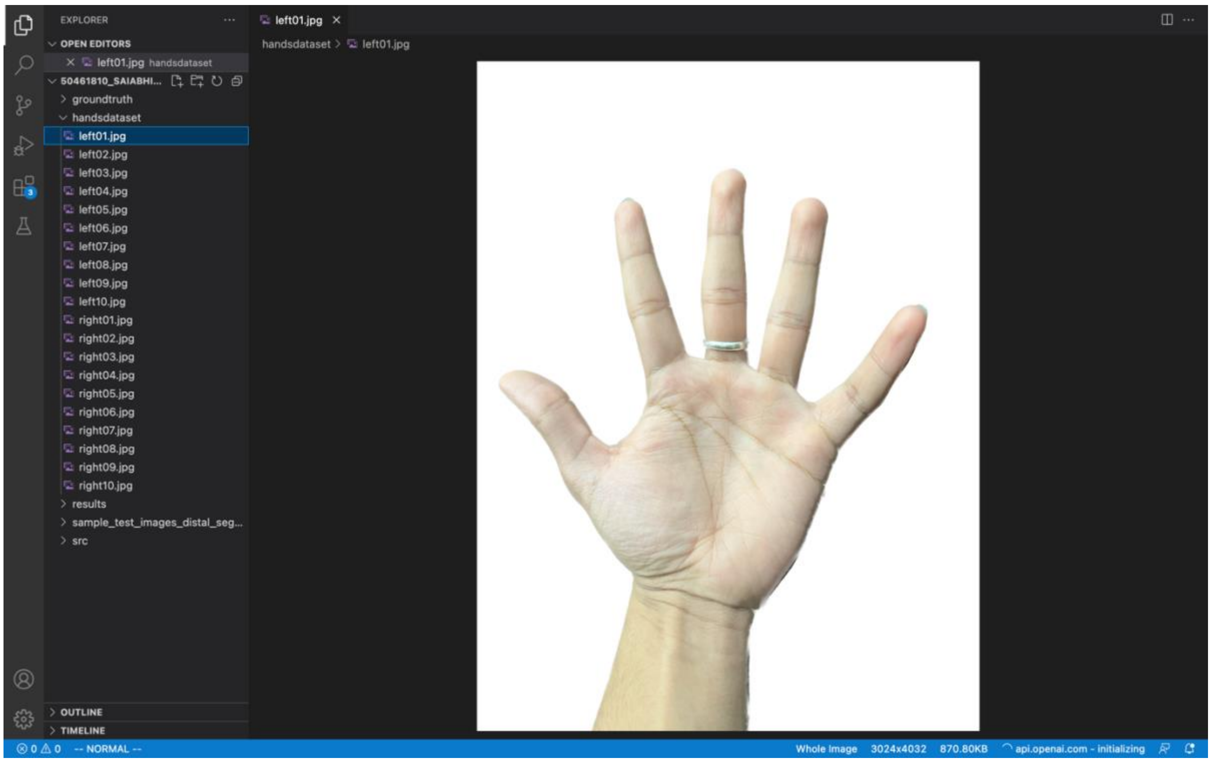
Name- Sai Abhinav Reddy Badinehal
UBIT Name- saiabhin
UB-PersonNo- 50461810

1. **Data collection (10 points)**: Collect a set of your own hand image (minimum 10 images for each hand) at multiple orientations and distances

Solution:

- I have taken 20 images (10 left, 10 right) with different orientations and distances using phone and saved them in the folder 'handsdataset'.
- These 20 images will be used as dataset to my algorithm.

Results:



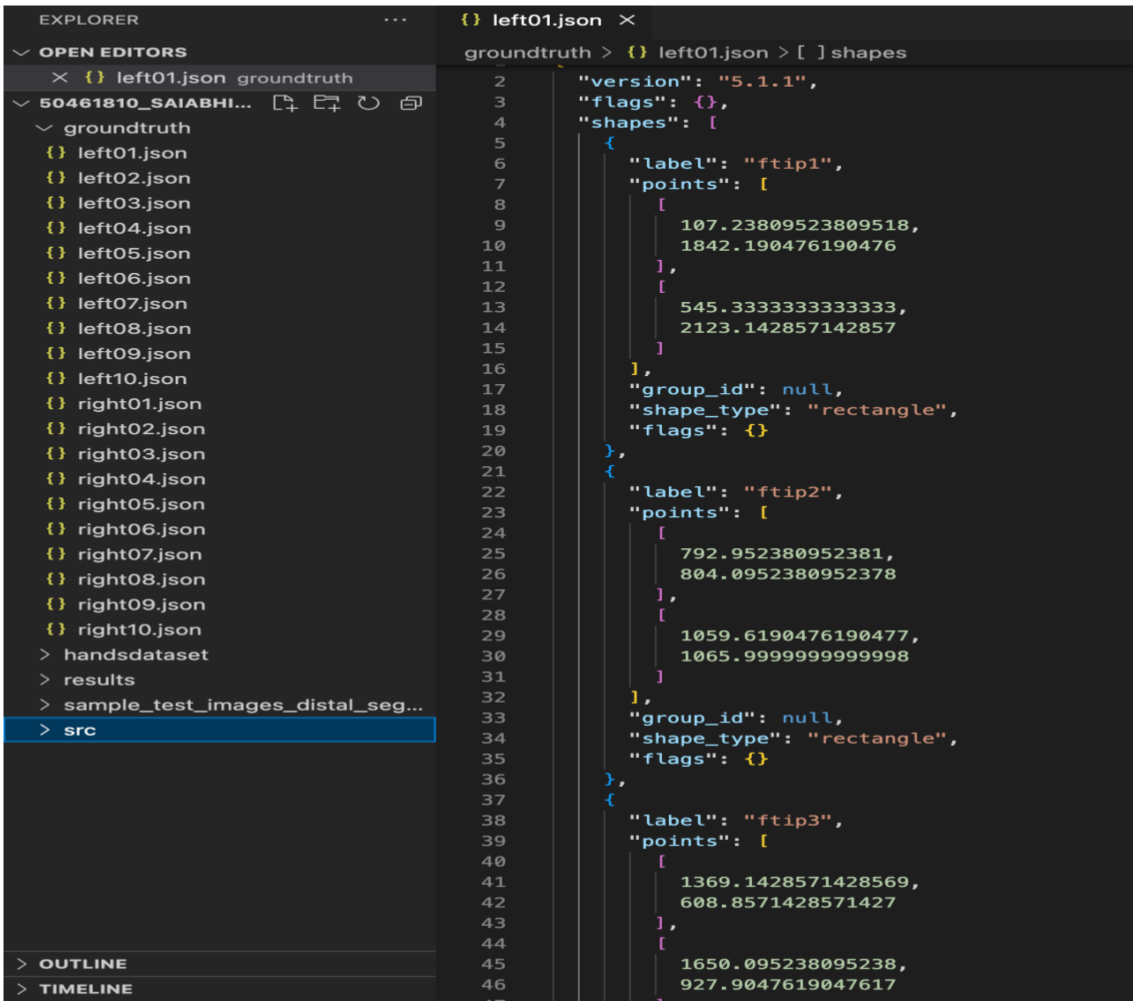
2. **Annotation (10 points):** Mark the box around each visible fingerprint region

Solution:

- For this problem, I have downloaded the labelme tool.
- Issued the following commands in my Mac
  - pip install labelme
  - python labelme
- Then using that tool I have annotated the 5 finger tips manually on all 20 hand images in 'handsdataset' folder.
- Finally I have saved the results in a json file in 'groundtruth' folder.

Results:

The results are displayed as follows:



```
EXPLORER
OPEN EDITORS
  × {} left01.json groundtruth
50461810_SAIABHI...
  groundtruth
    {} left01.json
    {} left02.json
    {} left03.json
    {} left04.json
    {} left05.json
    {} left06.json
    {} left07.json
    {} left08.json
    {} left09.json
    {} left10.json
    {} right01.json
    {} right02.json
    {} right03.json
    {} right04.json
    {} right05.json
    {} right06.json
    {} right07.json
    {} right08.json
    {} right09.json
    {} right10.json
  > handsdataset
  > results
  > sample_test_images_distal_seg...
  > src
> OUTLINE
> TIMELINE
```

```
groundtruth > {} left01.json > [ ] shapes
    "version": "5.1.1",
    "flags": {},
    "shapes": [
      {
        "label": "ftip1",
        "points": [
          [
            107.23809523809518,
            1842.190476190476
          ],
          [
            545.3333333333333,
            2123.142857142857
          ]
        ],
        "group_id": null,
        "shape_type": "rectangle",
        "flags": {}
      },
      {
        "label": "ftip2",
        "points": [
          [
            792.952380952381,
            804.0952380952378
          ],
          [
            1059.6190476190477,
            1065.9999999999998
          ]
        ],
        "group_id": null,
        "shape_type": "rectangle",
        "flags": {}
      },
      {
        "label": "ftip3",
        "points": [
          [
            1369.1428571428569,
            608.8571428571427
          ],
          [
            1650.095238095238,
            927.9047619047617
```

3. **Detection (35 points)**: Write an algorithm to detect automatically

Solution:

- Here for the fingertip detection, I have used 'Mediapipe' library to detect the fingertip.
- Mediapipe:
  - This is a open source software built by google to develop real time applications.
  - Here Mediapipe offers so many computer vision tasks. In that I am using hand landmark detection to detect the fingertips.
- Here in Hand landmark detection of mediapipe it takes hand as an input and marks the hand with 21 points.
- In those 21 points we want only 5 tips that points to the tips of the fingers.
- These 21 points are labelled with ID's. The ID's which we require are
  - 4 – Thumb Tip
  - 8 – Index finger Tip
  - 12 – Middle finger Tip
  - 16 – Ring Finger Tip
  - 20 - Little Finger Tip
- Giving input, the images in 'handsdatset' to mediapipe. It marks the 21 ID's i.e., hand landmarks in that images. Looping through the hand landmarks and ID's I want, I have drawn the bounding boxes around the 5 fingertips
- Finally, I have saved the results in 'results/myhands' folder.

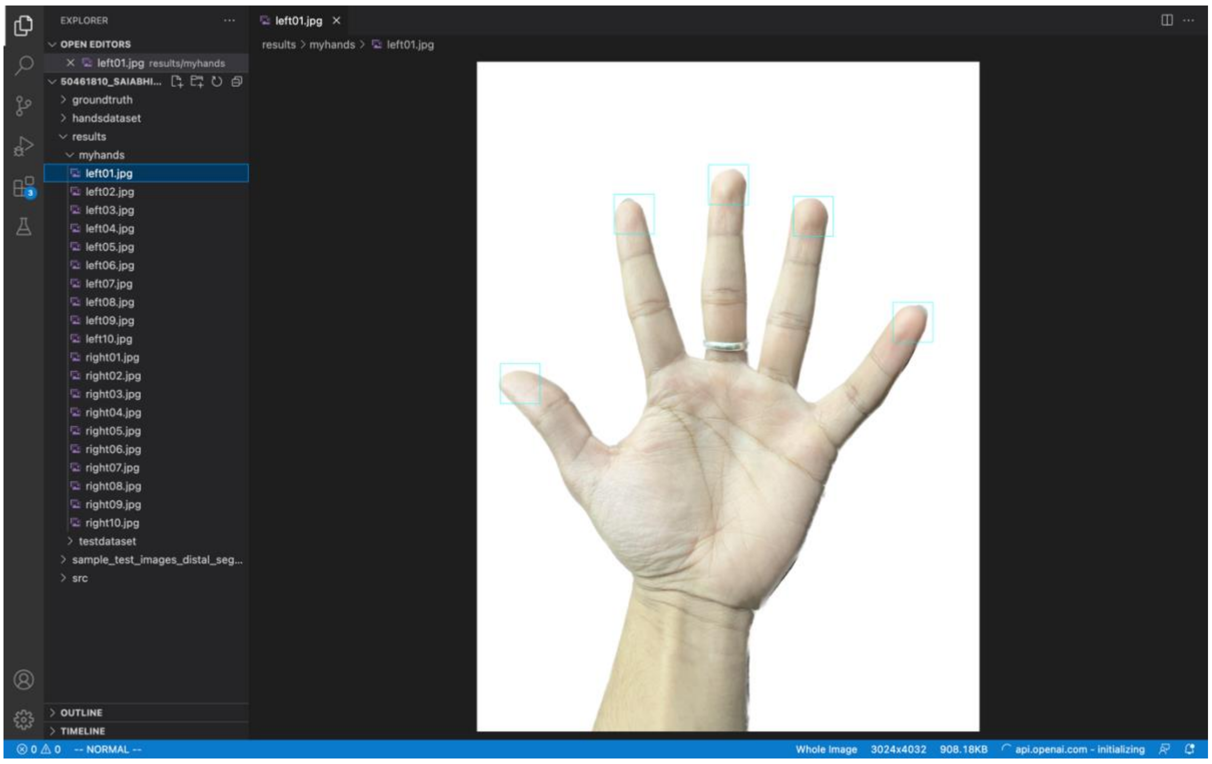
Execution:

Run the file 'main.py' in the 'src' folder. Make sure mediapipe library is installed. If not do it using:

'pip install mediapipe'.

This part of the code is written in function 'detect-fingerprint'. After detecting the fingertips I am storing the bounding box results in a list for the checking the performance of the Algorithm.

Results:



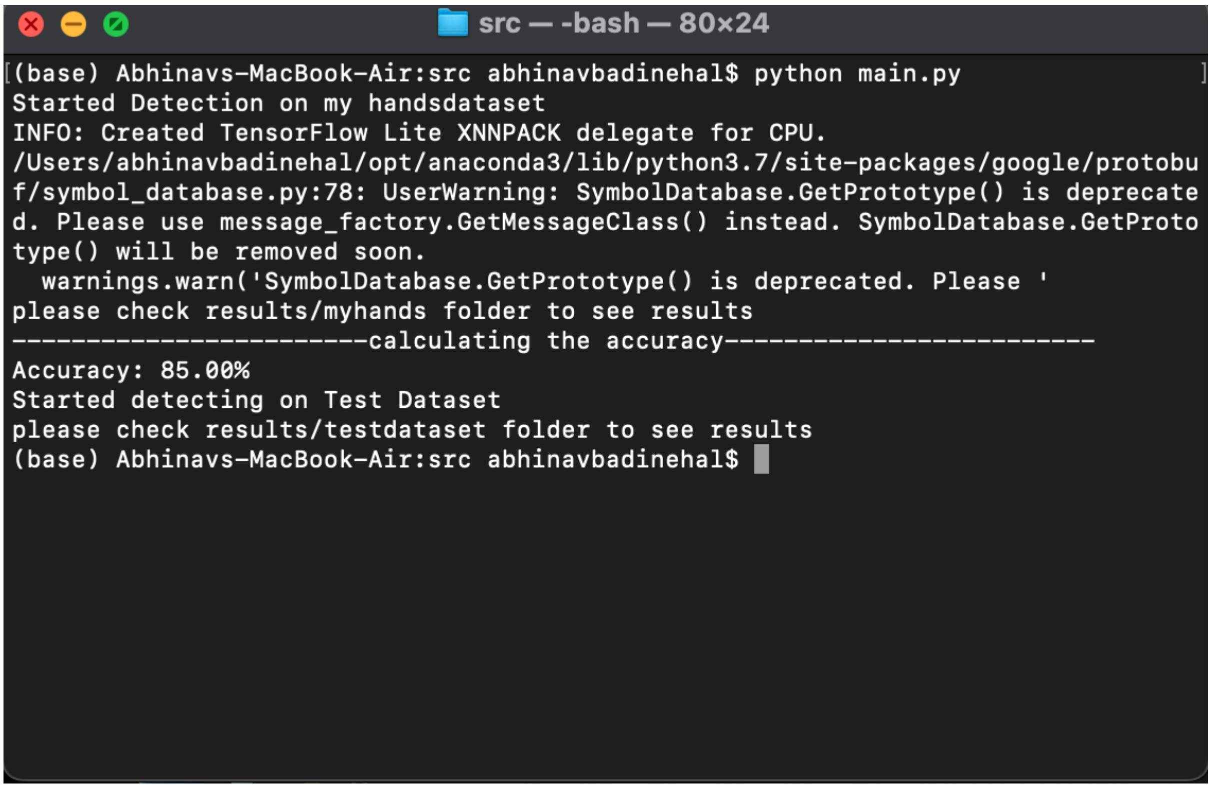
4. **Validation (10 points):** Evaluate your algorithm performance against the annotation

Solution:

- The steps I have done to evaluate the performance of my algorithm is
  - Created a function 'check_groundtruth' in main.py. So this function takes file name as input and get the manual annotations from the json file and store them in the list. This is called when detecting fingerprint for that particular image only.
  - Now I have 2 lists annotated list and predicted box list.
  - I am sending this 2 lists to calculate accuracy to 'get_accuracy' function.
- Next, for calculation accuracy we first calculate IOU so that we can know how many correct predictions are made by our algorithm.
- So calculate IOU for the same finger and store them in one list. Then calculate the Average IOU for all the 5 fingers.
- If that Average IOU is greater than IOU Threshold. Then the algorithm detected correctly.
- The Accuracy is no of correct predictions divided by total number of images.

Results:

My algorithm was able to achieve 85% accuracy against annotations which are done manually.



```
(base) Abhinavs-MacBook-Air:src abhinavbadinehal$ python main.py
Started Detection on my handsdataset
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
/Users/abhinavbadinehal/opt/anaconda3/lib/python3.7/site-packages/google/protobu
f/symbol_database.py:78: UserWarning: SymbolDatabase.GetPrototype() is deprecate
d. Please use message_factory.GetMessageClass() instead. SymbolDatabase.GetProto
type() will be removed soon.
  warnings.warn('SymbolDatabase.GetPrototype() is deprecated. Please '
please check results/myhands folder to see results
------------------------calculating the accuracy-------------------------
Accuracy: 85.00%
Started detecting on Test Dataset
please check results/testdataset folder to see results
(base) Abhinavs-MacBook-Air:src abhinavbadinehal$
```

5. **Testing (20 points):** Test your algorithm against the dataset provided.

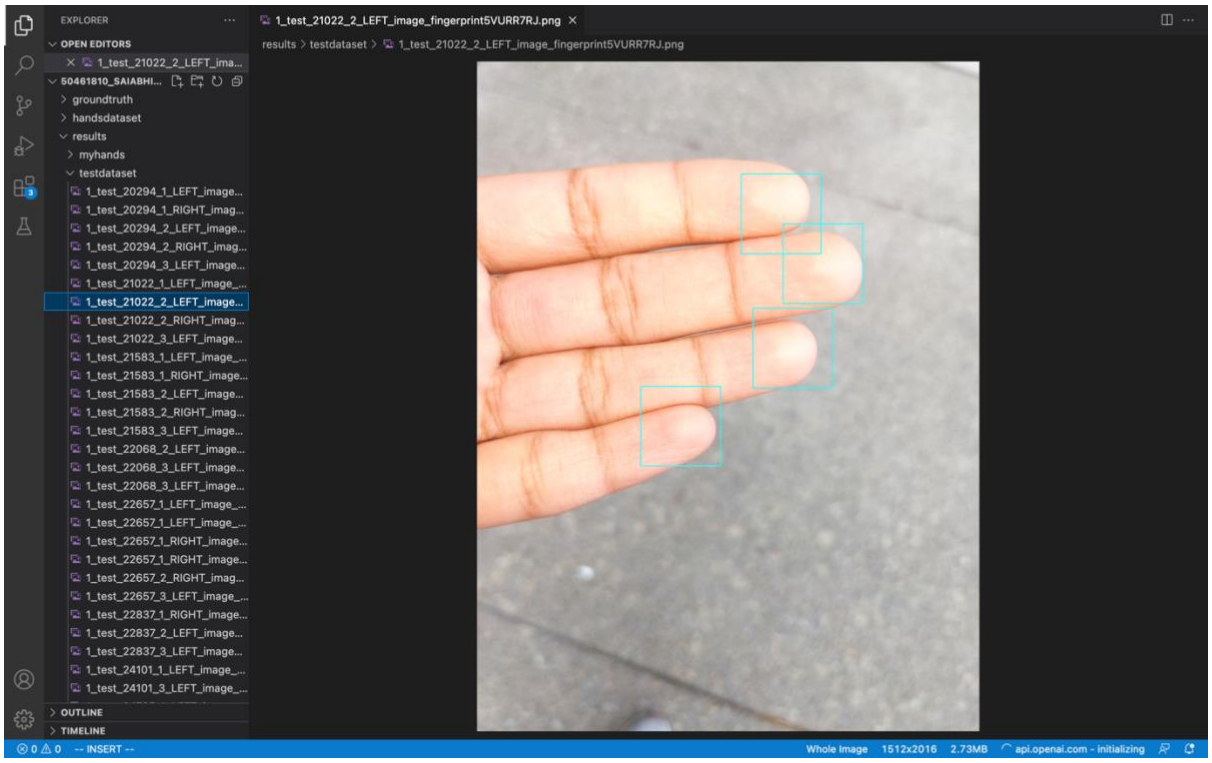
Solution:

- I have tested the Algorithm against dataset provided in the same ‘main.py’ file.
- Create a function ‘detect_fingerprint_test’ which detects the fingertips in the testdataset.
- Here in this function I have tuned in the parameters like min_detection_confidence and min_tracking_confidence by, telling the algorithm detect even though you didn’t find a whole hand and don’t have much confidence just detect the hand landmarks.
- The dataset images has only 4 fingers except thumb so I have changed required hand landmarks to class ID’s 8,12,16,20.
- Finally saving the results in the ‘results/testdataset’ folder.

Results:

Here the results are more biased towards the images that are bright and non blurry. The algorithm was able to detect fingertips if images are bright and non blurry.

Though some images are not clear, have more noise and dark out of 50 images it correctly detected 32 images which I think is enough to tell  my algorithm is performing well on this dataset.


EXPLORER
OPEN EDITORS
1_test_21022_2_LEFT_ima...
50461810_SAIABHI...
groundtruth
handsdataset
results
myhands
testdataset
1_test_21022_2_LEFT_image_fingerprint5VURR7RJ.png
results > testdataset > 1_test_21022_2_LEFT_image_fingerprint5VURR7RJ.png
OUTLINE
TIMELINE
-- INSERT --
Whole Image
1512x2016
2.73MB
api.openai.com - initializing

6. **Future scope (5 points):** How would you improve the detection performance?

Solution:

- Firstly, I would increase the no of images in the dataset.
- Secondly, I would give the model some high quality images to get itself more trained this is where test dataset is lacking.
- Thirdly, I would preprocess images removing background noises and filter them according to models input.
- Finally, I would tune in parameters to increase in the confidence of the model to improve detection performance.

## REFERENCES:

https://developers.google.com/mediapipe/solutions/vision/hand_landmarker

https://www.section.io/engineering-education/creating-a-finger-counter/

https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

https://www.youtube.com/watch?v=NZde8Xt78Iw&list=RDCMUCYUjYU5FveRAscQ8V21w81A&index=47