

CSE666- BIOMETRICS IMAGE ANALYSIS –

ASSIGNMENT-1 – REPORT

Name – Sai Abhinav Reddy Badinehal
UBIT Name – saiabhin
UB Person No - 50461810

1.(10 points) Annotation: Mark bounding boxes for each face in the image and save them using any tool.

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 faces manually from the count_faces.jpg image
- Finally I have saved the results in a json file 'count_faces_new.json' file in groundtruth folder.

Execution:

Please run '1_groundtruth.py' file which will set the ground truths for all the below problems which I used for evaluation. Basically I am training models here on annotated boxes to see how It performs on algorithm returned bounding boxes results.

Check for the paths of json files which must be changed accordingly. Then issue the command 'python 1_groundtruth.py' to see results.

Results :

The results in the file from labelme is displayed as follows.

```
{
  "version": "5.1.1",
  "flags": {},
  "shapes": [
    {
      "label": "face1",
      "points": [
        [
          1401.2727272727273,
          923.3766233766233
        ],
        [
          1445.4285714285716,
          976.6233766233767
        ]
      ],
      "group_id": null,
      "shape_type": "rectangle",
      "flags": {}
    },
  ],
}
```

2.(10 points) Face detection: Detect faces present in the image, label each region of interest (RoI) containing a face with a bounding box, and return the number of faces present in the image, i.e. the number of detected bounding boxes. Evaluate face detection models – (deterministic models and/or pretrained networks) against bounding boxes from step 1.

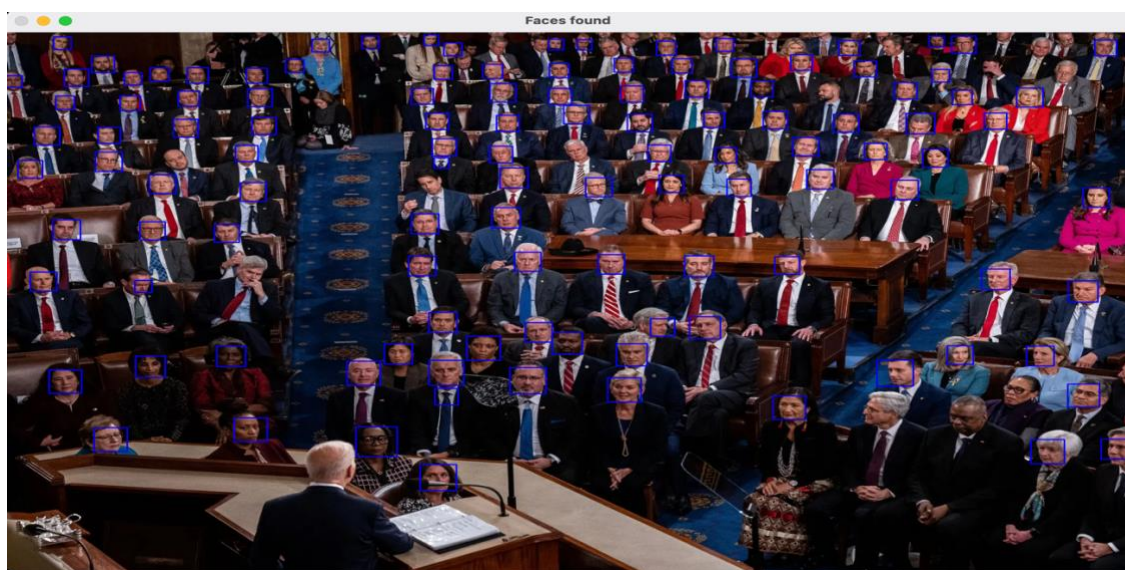
Solution:

- Here for the Face detection, I have used 'Haar Cascade-Classfier' to detect the faces.
- I have tuned in the detection scale to 1.12 and min Neighbour to 3 to get the best result from the given image.
- Then I have drawn the bounding boxes for the detected faces and stored the results in a json file.
- You can see the results in the 'results' folder 'boundingbox_results.json'.
- Also printed the number of faces detected.
- Evaluation:
 - Normally, if we see the face detection model will be evaluated on metrics such as 'IOU,precision,recall,F-1score'. Here I have evaluated the IOU.
 - Here I have my ground truth and bounding box results, as we know we cannot compare them based on their labels as the order I annotated and order achieved algorithmically different.
 - So what I am trying to do is for every ground truth bounding box I am finding the best predicted bounding box where I can get iou higher and then computing it for true positives, false positives and false negatives so I can get precision, recall and F-1score.
 - As the results are attached I have got my precision 0.95, Recall 0.83 and precision of 0.88. Therefore my model is performing well.

Execution:

Once navigate to file '2_facedetection.py' and check the path of images and json results. Change it according to your directory while execution. Then issue the command 'python 2_facedetection.py' to see results.

Results:



```

(base) Abhinavs-MacBook-Air:src abhinavbadinehal$ python 2_facedetection.py
No of persons present are 117
-----Evaluations-----
Precision is: 0.9576271186440678
Recall is : 0.8308823529411765
F1-score: 0.8897637795275591

```

3.(10 points) Sentiment/Expression analysis: For each identified RoI, analyze the expression to return metrics like anger/disgust/sadness, etc. Evaluate the performance of your model.

Solution:

- For the Facial Expression detection, The library I used is 'DeepFace' library.
- I have taken the bounding box results from step2 by importing data from json file.
- Captured the faces from those values and sent them to DeepFace.analyze to detect the emotions.
- It was able to give the results with top emotion of that particular detected face.
- Evaluation:
 - As I have mentioned at first by running '1_groundtruth.py' training is done on faces which I manually annotated which is considered as ground truth for this problem.
 - Similarly what we have done for face detection finding the best iou and main step is if its above threshold I am checking if the both of emotions matches.
 - By this I have found precision, recall and F1-score and obtained results as 0.91,0.71,0.80 respectively.
 - By this I can say my model is performing well.

Execution:

Once navigate to file '3_facialexp.py' and check the path of images and json results. Change it according to your directory while execution. Then issue the command 'python 3_facialexp.py' to see results.

Results:



```
-----Evaluations-----
Precision is:  0.9137931034482759
Recall is : 0.7162162162162162
F1-score: 0.8030303030303031
```

4.(10 points) Gender: Classify each detected face by gender. Evaluate the performance of your model against gender information from the dataset.

Use information from the following resources to evaluate your results:

<https://cawpdata.rutgers.edu/women-elected-officials/position?current=1&position%5b%5d=US+Senator>

<https://cawp.rutgers.edu/facts/levels-office/congress/women-us-congress-2023>

For the purpose of this exercise, assume all congresspeople not listed in the above resources as male.

Solution:

- Here, for the Gender Detection. I have used 'Deep Face' library.
- I have taken the bounding box results from step2 by importing data from json file.
- Captured the faces from those values and sent them to DeepFace.analyze method to detect the gender.
- It was able to give the results with dominant gender of that particular detected face
- Evaluation:
 - As I have mentioned at first by running '1_groundtruth.py' training is done on faces which I manually annotated which is considered as ground truth for this problem.
 - Similarly what we have done for face detection finding the best iou and main step is if its above threshold I am checking if the both of gender matches.
 - By this I have found precision, recall and F1-score and obtained results as 0.95,0.83,0.88 respectively.
 - From the resources given there are 28% of women. If my model is somewhat accurately predicting to that proportion then it is good model. From the results I can say my model is performing good if I compare with above resources information visually.

Execution:

Once navigate to file '4_genderanalysis.py' and check the path of images and json results. Change it according to your directory while execution. Then issue the command 'python 4_genderanalysis.py' to see results.

Results:



```
-----Evaluations-----
Precision is:  0.9545454545454546
Recall is : 0.8333333333333334
F1-score: 0.8898305084745762
```

5.(10 points) Face pose estimation: For each identified RoI, estimate face pose to determine whether the attendee is looking straight or to a side. Evaluate the performance of your model.

Solution:

- Here, for the Face Pose Estimation. I have used 'Head Pose' library.
- I have taken the bounding box results from step2 by importing data from json file.
- Captured the faces from those values and sent them to PoseEstimator.pose_from_image() method to get yaw, pitch, and roll values.
- From those values I have described if person is looking sideways or straight.
- Evaluation:
 - When I am trying to train the groundtruth with headpose library. It was not able to detect the faces may be due to small faces and lighting. Therefore its more pixalated.
 - After going through many articles based on the values I have obtained for yaw, pitch, roll I can tell my model is working good by looking at it results.

Execution:

Once navigate to file '5_facepose.py' and check the path of images and json results. Change it according to your directory while execution. Then issue the command 'python 5_facepose.py' to see results.

Results:



6.(10 points) Feature extraction: Extract features for each face detected using pretrained models and store the generated embeddings for matching.

Solution:

- For the Feature Extraction, The library I used is 'face_recognition'.
- From face_recognition I have loaded the image and found the face locations.
- From those face_locations I have got the face_encodings and stored them in embeddings.npz file in the results directory.
- Where this embeddings can be used in recognizing faces in the further question.

Execution:

Once navigate to file '6_featureExtraction.py' and check the path of images and json results. Change it according to your directory while execution. Then issue the command 'python 6_featureExtraction.py'

Results:

Navigate to results folder and then embeddings.npz file to see the results.

7.(25 points) Face recognition: Given a dataset containing images and names of lawmakers, identify each RoI and return a label with the Senator/Congressperson's name. (Note that not all attendees present will be included in the dataset since not all attendees are lawmakers. For such cases your solution must return the label = "Unknown"). Evaluate the performance of your matcher.

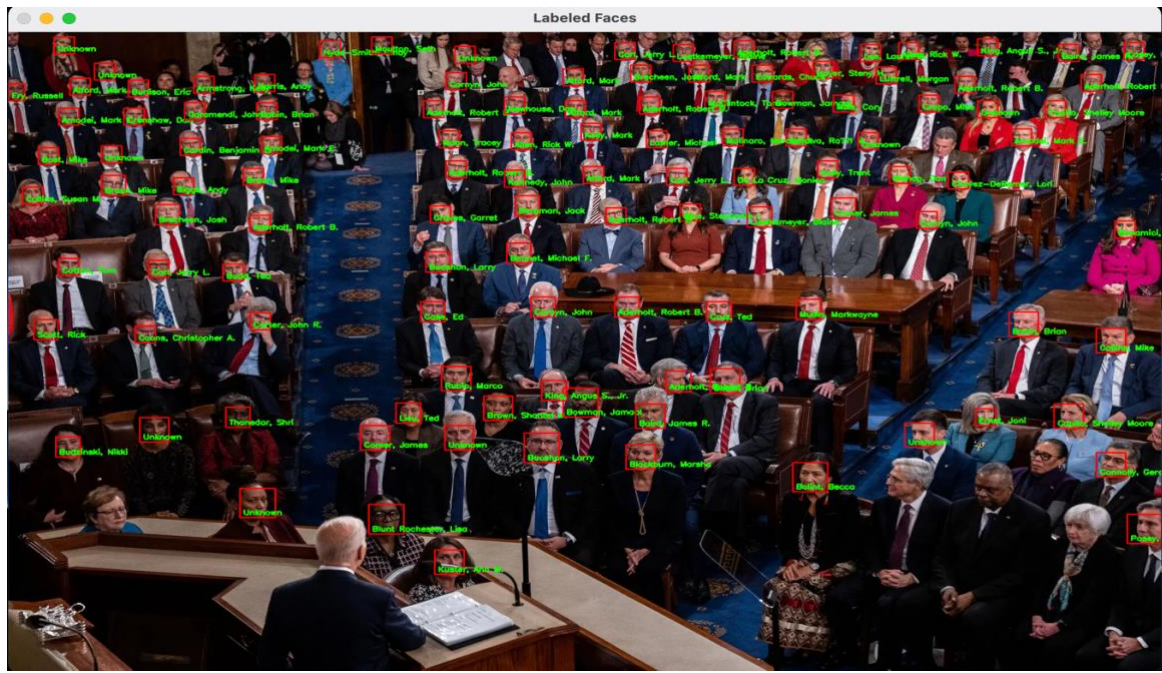
Solution:

- Here, I am looping through dataset and .tsv file and storing the names and file paths in a dictionary.
- Now from the file paths I am trying to load each image and finding their encodings and storing them in dictionary as well. This saves us lot of time.
- From the input biden image generated encodings from previous problem I am getting the encodings and comparing them with dataset encodings which I stored in the dictionary.
- If I found the match, I am returning the label of the person with his name else "Unknown".
- Evaluation:
 - When I am trying to train the groundtruth with face_encodings I am getting null encodings. May be due to small faces and more pixalation. Therefore I am not able to find encodings to ground truth bounding box.
 - So I have manually evaluated its matching it was able to match most of the faces accurately.

Execution:

Once navigate to file '7_facerecognition.py' and check the path of images and datasets .tsv files. Change it according to your directory while execution. Then issue the command 'python 7_facerecognition.py'

Results:



REFERENCES:

<https://github.com/wkentaro/labelme/blob/main/README.md>

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

<https://pypi.org/project/deepface/>

<https://pypi.org/project/headpose/>

https://face-recognition.readthedocs.io/en/latest/face_recognition.html

<https://support.pix4d.com/hc/en-us/articles/202558969-Yaw-Pitch-Roll-and-Omega-Phi-Kappa-angles/>