SoC Summer Workshop

Project

July 2021

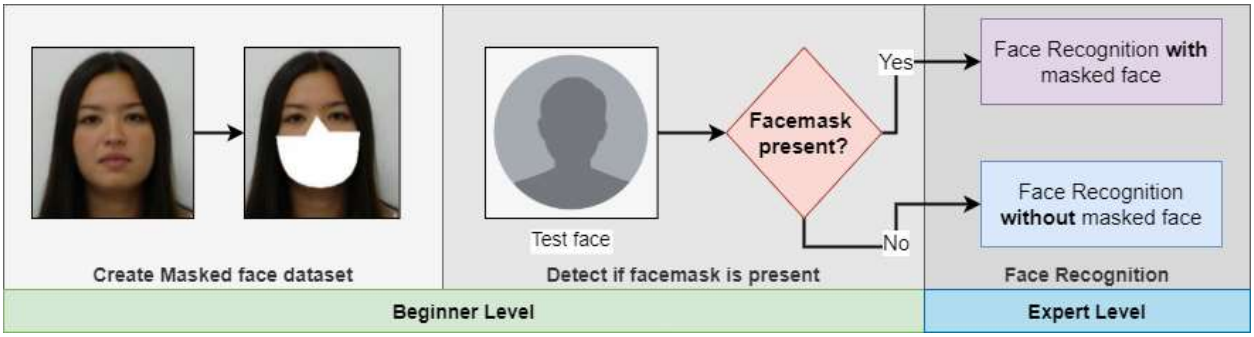
Masked Unmasked Face Recognition

Face Recognition is one of the most popular biometric authentication methods in the present days. As more people are wearing masks during the Covid-19 pandemic it is hard for the face recognition methods to function as intended since the face is now only partially visible. This posses a problem in many scenarios where face recognition is used regularly. For example, we can think of a border crossing area where face recognition is installed.

In this project, you are required to build the followings:

1. **Create a masked face dataset**: Due to the lack of public dataset with masked faces, you would be creating a synthetic dataset from a given face (unmasked) dataset.
2. **A Facemask detector**: Given a face image, this would detect if the person in the image is wearing a facemask or not.
3. **Face recognition model for unmasked faces**: This face recognition model will authenticate people who are NOT wearing any mask.
4. **Face recognition model for masked faces**: Like the previous one, this model will authenticate people who are wearing facemasks.

This project is a great opportunity to learn the basic of face recognition methods and how to tackle new challenges with computer vision applications.



Beginner Level:

We use the Georgia Tech Face database which has contains images of 50 people taken at resolution 640x480 pixels. The average size of the faces in these images is 150x150 pixels. Images of each subject reside in separate folders named "sXX" where "XX" is the number of the subject (from "01" to "50"). Each such folder contains 15 images of a subject, named from "01.jpg" to "15.jpg".

The dataset is available as a zip file in your project folder under the name "Dataset_1.zip".

For the beginner level, we make use of the "starter.py" code, which you can find in the project directory. Follow along with the tasks and fill in the blanks of the given code to complete beginner level.

Tasks:

T1: Reading images.

- Change the `dataset_path` to point to the unzipped FaceDataset_1/ folder in your computer.
- The given outer loop will go through all the sub-folders in the folder. The inner loop will go through all the images of a sub-folder (each sub-folder belongs to one subject).
- Complete the code to read the images.
- The labels for each image have been already appended to list y for you.

T2: Pre-processing images.

- In the given loops (outer and inner) use the dlib's face shape detector library (sample) with the pretrained model (download) to detect face in an image. This would help you to crop out the image in the next step.
- Use the detected face keypoints from the previous step to crop the face out of the larger image. Now resize the image to 150x150 pixels.
- Convert the cropped and resized image from the previous step to grayscale.
- Complete the rest of the codes to mark the end of this task.

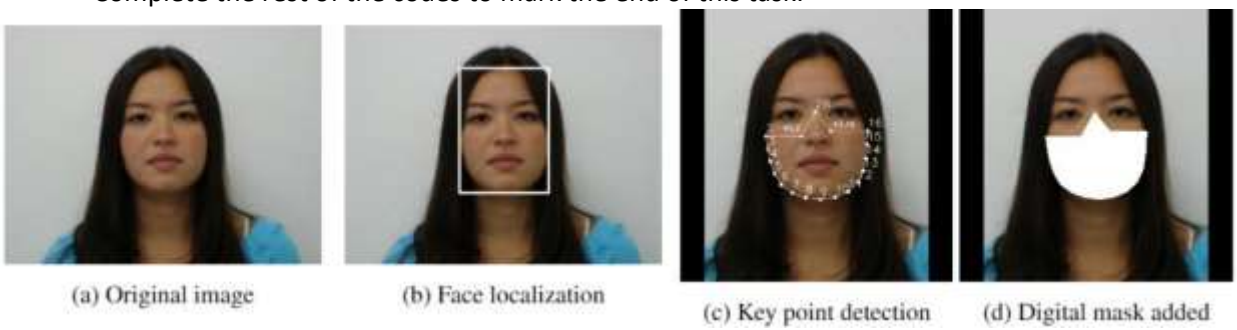


Figure: Process of adding digital (synthetic) mask to a face image

T3: Create masked face dataset.

- Like the previous task, for each image, detect face keypoints in the image.
- You can use a subset of the keypoints to draw facemask on the image using cv2.fillPoly method.
- After the end of this task, `X_masked` will contain all the images with added digital/synthetic facemask.
- Refer to the image at the bottom of page 2 for the processes to follow.

T4: Mask detector.

- Prepare features using the given code (already done for you).
- Write code to split `X_processed, X_masked` and `y` into training and testing sets. Make use of the *"`sklearn.model_selection.train_test_split`"* to do this. Use an 80-20 split and make sure to shuffle the samples.
- Use the sklearn SVM package to train a binary classifier using `x_train` and `y_train`. The classifier should output if a given image has facemask, or not.
- Use the `x_test` and `y_test` to evaluate the classifier and print the accuracy value.

Expert Level:

We will build upon the beginner level code to try out different techniques and improve our model. The same dataset will be used here.

Complete the following tasks,

Tasks:

T1: Different pre-processing techniques

- What are other pre-processing steps you can use?
- Examples: Keep 3 channels (RGB), add a gaussian blur to reduce noise, etc.
- Try few other pre-processing techniques and evaluate how they affect accuracy.

T2: Train a Face Recognition model for faces WITHOUT mask.

- You can train a face recognizer of your own choice for the 50 subjects. Train the model to learn to approve any face from 30 subjects out of the 50. Use the rest 20 subjects as negative test subjects. Use the X_processed (not X_masked) list as X_processed contains only the images WITHOUT facemask.
- The face recognizer should take a probe image as an input, output "ACCEPT" when the probe images is of a subject from the 30 which the system/model recognizes. And the model should output "REJECTED" when a face out the 30 subjects is presented to the model.
- For a start you can use the face_recognition package (link).
- Report on the accuracy of the model.

T3: Train a Face Recognition model for faces WITH mask.

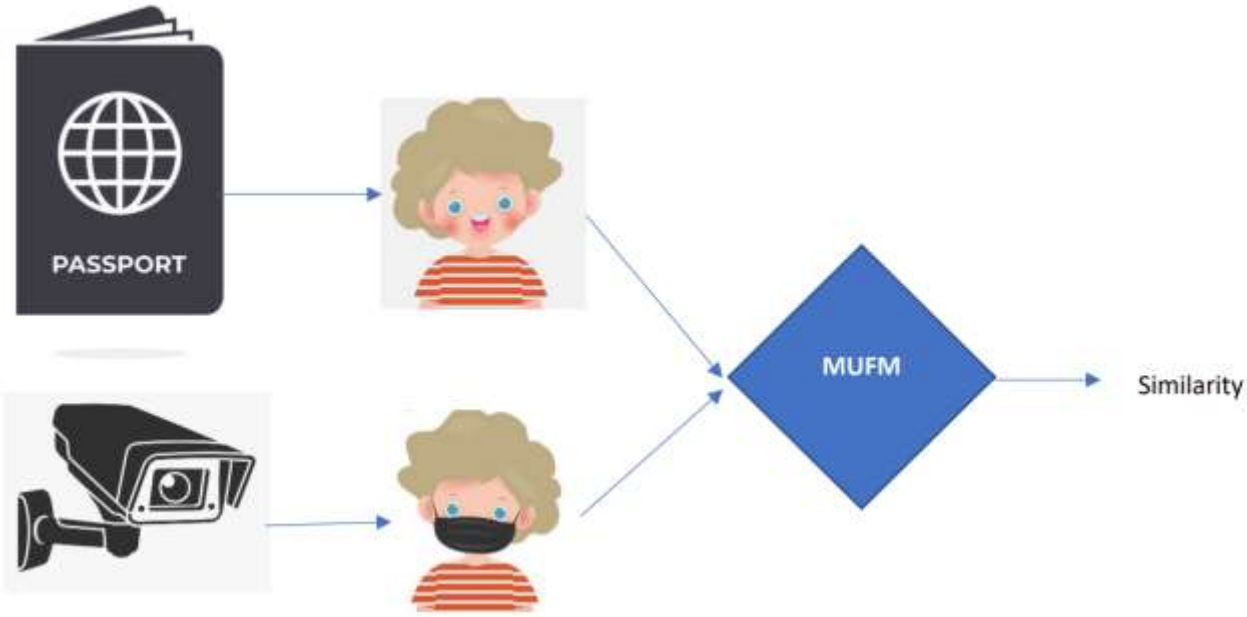
- This is exactly similar to T2. The only difference is that your training images for the model will be face WITH masks (use the X_masked list this time).
- Other details are same as T2.

T4: Combine the pipeline with the tasks from Beginner level.

- Combine these tasks to complete the pipeline as shown in the figure at the top of page 2.

Bonus Level:

In this level we will create a face recognition model (Masked Unmasked Face Matching - MUFM) which can recognize a probe face image of a known subject WITH mask, even when the model was trained using only face images WITHOUT facemasks. See the picture below to get a better idea.



First, use the same dataset (Georgia Tech Face database) as the previous tasks and report on the model accuracy. Next, use the larger celebA face dataset. Process of downloading the celebA dataset is given below:

1. Go to the google drive link https://drive.google.com/drive/folders/0B7EVK8r0v71pTUZsaXdaSnZBZzg
2. Go inside `img_align_celeba_png` folder
3. Download `img_align_celeba_png.7z001 - img_align_celeba_png.7z016` all files
4. Combine them and unzip it using 7zip
5. Unzip the file to `celeba/images` folder
6. Download annotations `identity_CelebA.txt from` the following link https://drive.google.com/drive/u/1/folders/0B7EVK8r0v71pOC0wOVZlQnFfaGs?resourcekey=0-pEjrQoTrlbjZJO2UL8K_WQ
7. Place annotations `identity_CelebA.txt` under `celeba/`