Advanced Artificial Facemask Overlay

Workspace layout

```
├── advanced_artificial_facemask_overlay:
│   │   main.py
│   │   shape_predictor_68_face_landmarks.dat
│   ├── DATASET
│   │   ├── cover_both
│   │   ├── slice_face
│   │   ├── uncover_both
│   │   ├── uncover_nose
│   ├── face
│   ├── mask
```

The “face” and “mask” folders are the default locations for unprocessed face and mask images, as well as mask landmark CSV files. The “DATASET” directory is the parent folder for saving different categories of output images images, which includes sliced faces removed from the rest of the image (slice_face), face with mask overlaid by only covering the chin (uncover_nose), uncovering nose (uncover_nose), and correct covering face (cover_both).

The main script takes the following parameters:

-mp or **--maskfolder** Absolute path to mask folder

-fp or **--facefolder** Absolute path to face folder

-dp or **--datasetfolder** Absolute path to face folder

-p or **--PATTERN** choices: cover_both, uncover_both, uncover_nose, slice_face
Overlay mask with "slice_face", "uncover_nose", "uncover_both" or "cover_both" options

Facial landmark detection

By using the face landmark detector, mask images can be accurately and naturally overlaid on the target face image. “shape_predictor_68_face_landmarks.dat” is the necessary dependency file for the **dlib** face landmark detector. It detects 68 feature points on a person’s face, an example is shown below.

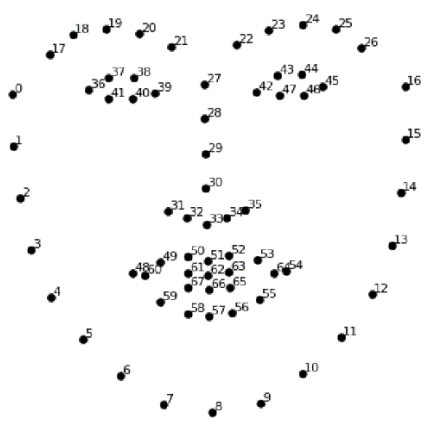

Figure 1-dlib 68 face landmarks map

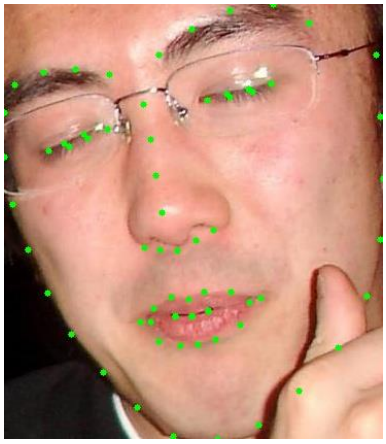

Figure 2-The landmarks detected on human face

The figure 1 is the point map of facial landmarks. And on the right is the result of applying the detector to the face image and drawing green points on the landmarks.

Mask images and landmarks

The mask landmarks are marked by a webapp called Make Sense, and should consists of 12 points with landmark numbers 1, 4-12, 15 and 30. These numbers are matched to the face landmark points as shown in figure 3 and 4. This landmark data needs to be exported as a csv file with the same file name as the mask.

Be aware that the mask must be a PNG with a transparent background/alpha channel for the optimal overlay effect.

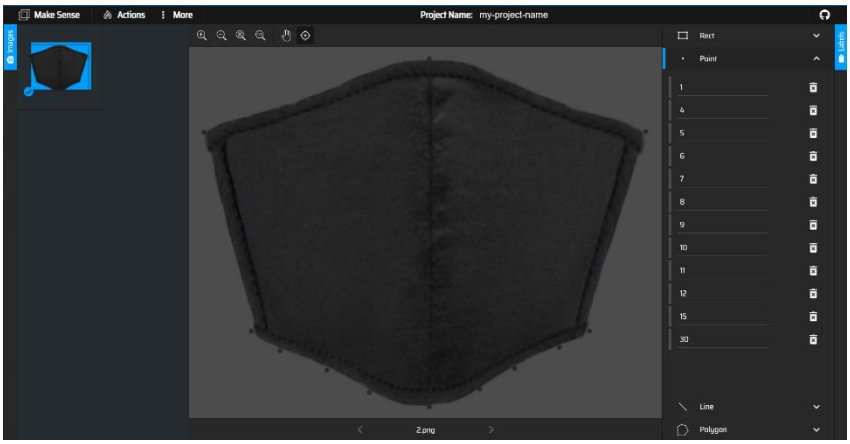

Figure 3-Maring landmarks for the mask with Make Sense

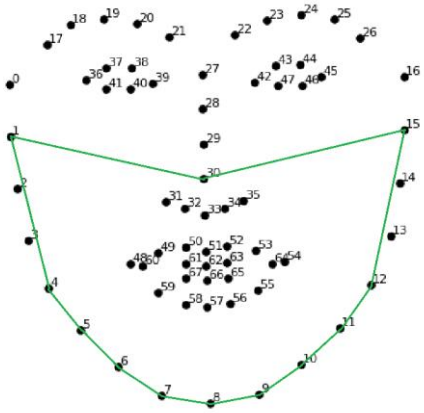

Figure 4-12 mask with corresponding face landmarks

It is possible to generate the mask landmark points with any marking tool as long as it is in the following format:

```
landmark_number,x_coordinate,y_coordinate,file_name,image_width,image_height
1,14,67,1.png,489,415
4,0,220,1.png,489,415
…
15,471,67,1.png,489,415
30,246,0,1.png,489,415
```

With correct mask images and csv files contain the landmark data, the mask folder should look like figure 5.

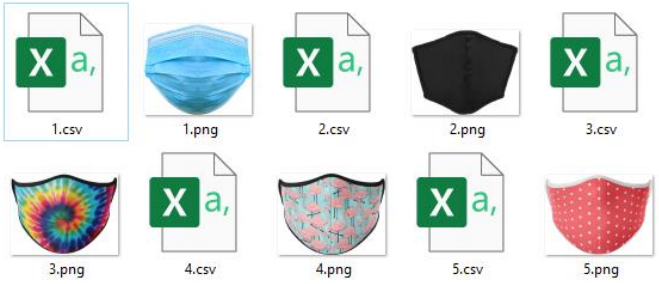

Figure 5-Example mask folder

Things to note

For performance reasons, only images with one face can be processed, otherwise it will be ignored. To create a versatile mask detector model, a range of different face attributes should be used when generating different types of artificial dataset images with a variety of “PATTERN” choices. Please be aware that as the resolution of the image increases, the longer the processing time.

Be aware that only JPG images are accepted.

Dataset with different categories

The 4 patterns that can be chosen are as follows:

```
cover_both, uncover_both, uncover_nose, slice_face
```

Corresponding results (for comparison the same image is used):

Figure 6-Different categories of example result

The first image shows the face landmark points in green and the mask landmark points in yellow.

As you can see by applying a homography matrix to the mask image using the facial landmarks as target points, we can manipulate the mask to cover any section of the face we want.