A Appendix

OpenFilter: implementation details

Most of the AR filters available on social media platforms can only be applied in real-time on selfie images captured from the camera. Hence, it is challenging to carry out quantitative and systematic research on such filters. OPENFILTER fulfills such a need through (1) an Android Emulator, (2) a computer and (3) a virtual webcam. Through an *auto-clicker* system, each image is first projected on the camera; next, the filter is applied to the image and finally the filtered image is saved on disk.

For the auto-clicker to work, it is necessary to place the required elements in a precise position on the desktop. More details are available in our repository, and an exemplary screenshot can be found in Figure 5. Note that OPENFILTER saves the filtered images by taking a screenshot, rather than downloading the image directly from the social media app. This is motivated by the will of accelerating the process: very often, images treated with AR filters are downloaded as videos, causing remarkable delays. OPENFILTER is designed to filter large collections of images and, as a consequence, the fluidity of the system is one of the main specification requirements.

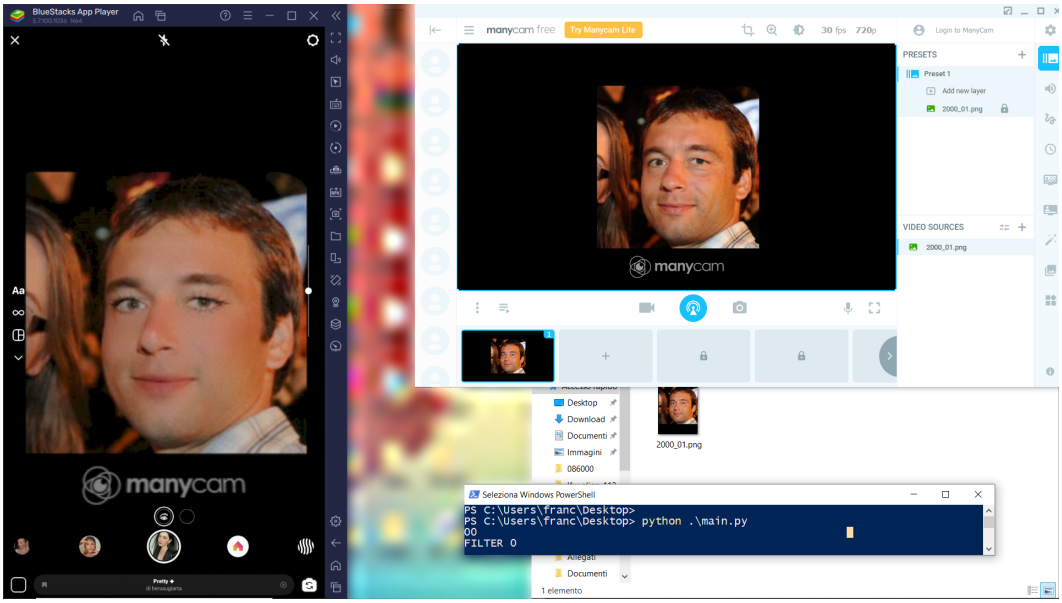

Figure 5: Screen set-up for OPENFILTER. Android simulator on the left, virtual camera on the right. The image is projected on the camera opened on Instagram and the selected filter is directly applied on the image.

Next, we highlight several code snippets of OPENFILTER. In listing 1, we include the requirements. Note that since we are dealing with an auto-clicker, we identify the positions of the different elements on the desktop. The values reported correspond to a full-HD desktop (1920 x 1080 pixels), and will need to be re-calibrated in case of different screen resolutions. In particular, `new_file` refers to the position of the next image that will be processed by the system; `many_cam` and `many_cam_confirm` respectively refer to the area on the virtual camera where the new image is dragged and the confirmation button on its interface; `screen` refers to the four corners of the area where the screenshot is taken to save the filtered image; `right_filter` and `left_filter` respectively refer to the position of the next filter on the right and on the left of the current one.

In listing 2, we include the relevant functions that are implemented in the auto-clicker. The function `drag_and_drop` is used to move the images on the desktop, processing them sequentially. The calls to the function `sleep` are calibrated to the response times of software on a Intel(R) Core(TM) i7-8565U machine with NVIDIA GeForce MX150. The functions `h_padding` and `v_padding` create a padding around the target image, so that the buttons of the interface of the social media app (Instagram in our case, see Figure 5) do not overlap with the image. The full code including the order of the different actions performed by the auto-clicker is available in our repository.

Listing 1 Requirements

```
import random
import time
import numpy as np
import os
import argparse
from PIL import Image

import pyautogui as auto

positions = {
    "new_file": (1220,750),
    "many_cam": (1150,250),
    "many_cam_confirm": (1180,280),
    "screen": (42,305,510,510),
    "right_filter": (430,980),
    "left_filter": (150,980)
}
```

Listing 2 Functions

```
def drag_and_drop(start, end):
    auto.mouseDown(start)
    auto.moveTo(end)
    auto.mouseUp()

def sleep(ms):
    seconds = ms / 1000
    seconds += random.random()*seconds/10
    time.sleep(seconds)

def h_padding(img):
    background = Image.new('RGB', (img.size[0] + 5*img.size[0]//285, img.size[1]))
    background.paste(img, (background.size[0] - img.size[0], 0))
    return background

def v_padding(img):
    background = Image.new('RGB', (img.size[0], img.size[1] + 150*img.size[1]//285))
    background.paste(img, (0, (background.size[1] - img.size[1])//2))
    return background
```

Dataset Documentation

We have beautified two face datasets using OPENFILTER which we also share in this contribution.

FAIRBEAUTY is a beautified version of the FAIRFACE dataset, following the same nomenclature for the files and the same dataset documentation.[14] FAIRFACE is publicly available with a CC BY 4.0 license. This license enables sharing, copying and redistributing the material in any medium and format and adapt, remix, transform and build upon the material for any purpose, even commercially. Hence, we had permission to create the FAIRBEAUTY dataset as derivative work. We share FAIRBEAUTY with a CC-BY-NC-SA 4.0 license which does not allow the use of the datasets for commercial purposes.

In the case of FAIRBEAUTY, the original folders (train and validation) are divided into subfolders according to the name of the images (e.g. images from 0_01.*png* to 999_01.*png* are in subfolder 000000, from 1000_01.*png* to 1999_01.*png* in subfolder 001000, and so on). Additionally, we

[14] More information available on the official Github repository of FAIRFACE: https://github.com/joojs/fairface.

provide metadata regarding the filter that is applied on the images. This information is enclosed in the files `filters.txt` that can be found in the two main folders (train and validation). These files associate the filter name to the name of a subfolder. All the images in a subfolder are beautified using the same filter. An extract of `filters.txt` is shown in listing 3: on the left-hand side we provide the name of the subfolder, and on the right-hand side the name of the utilized filter.

Listing 3 Filters.txt

```
021000: big city life
022000: hary beauty
023000: Pretty
024000: Shiny foxy
025000: Just Baby
026000: hary beauty
027000: Pretty
```

To facilitate the access to the files, we provide a synthetic representation of the filenames in FAIRBEAUTY through the regular expressions available in listing 4, distinguished for the two different folders (train and validation).

Listing 4 Regular expressions for FAIRBEAUTY files.

```
train_fair_beauty
- ^0[0-86]{2}000$
- - ^[0-9]{1,5}_0[0-9]{1}\.png$
- filters.txt

val_fair_beauty
- ^0[0-10]{2}000$
- - ^[0-9]{1,5}_0[0-9]{1}\.png$
- filters.txt
```

B-LFW is a beautified version of the LFW (Labeled Faces in the Wild) dataset, a public benchmark dataset for unconstrained face recognition. In particular, we beautify the aligned version with the images rescaled at 112x112 pixels[15], where the images are shared as a `carray` of the `bcolz` Python library. For the beautification purposes, we extract the images from the array and convert them to png files. The nomenclature of the files in our dataset follows the index of the images in the original array (i.e. the entry 0 of the array is coverted to $0.png$). In the case of B-LFW, the filters correspondence for every image is synthesized into the `numpy` array `filters.npy`. The entries of this array are integer number from 0 to 7, referring to the eight beauty filters. The position of the filter ID in the array corresponds to the name of the beautified image (i.e. if entry 0 of `filters.npy` is equal to 2, then filter 2 is applied on image $0.png$). Please refer to listing 5 for the correspondence between filter IDs and names.

Listing 5 Correspondence between filter IDs and names.

```
filter 0: "pretty" by herusugiarta
filter 1: "hari beauty" by hariani
filter 2: "Just Baby" by blondinochkavika
filter 3: "Shiny Foxy" by sasha_soul_art
filter 4: "Caramel Macchiato" by sasha_soul_art
filter 5: "Cute baby face" by sasha_soul_art
filter 6: "Baby_cute_face_" by anya__ilichev
filter 7: "big city life" by triutra
```

The regular expressions for the nomenclature of the files in B-LFW are available in listing 6. Note that, aside from B-LFW, we also share eight different versions of the LFW dataset: in each version,

[15] Version available at: https://github.com/ZhaoJ9014/face.evoLVe.

all the images are beautified with one of our selected filters. This allows reproducing our experiments, and investigating each filter separately.

Listing 6 Regular expressions for B-LFW files.

```
lfw_align_112_png_beauty
- ^[0-7]{2}_[0-7]{2}$
- - ^[0-9]{6}\.png$
- filters.npy
```

In B-LFW, the eight filters are applied in equal proportions on images in the dataset. In particular, different images from the same individual are beautified with different filters. For research purposes, we also share other eight versions of the dataset, in which all the images are treated with a specific filter. This allows reproducing the results shown in Experiment 2 and performing specific analyses on each filter separately.

Choice of the filters

Hundreds of "beautification" filters created by Instagram users are available on social media platforms. Unfortunately, there is no structured repository of all the available filters and there is no visibility regarding their popularity. Thus, to select a representative sample of filters, we had to rely on information provided by external sources –such as magazine reports or online articles featuring the filters– and/or on filters created by influential, digital filter creators on Instagram with thousands of followers. Our goal was to capture a representative sample of current beautification filters, trying to mitigate the unavoidable sampling bias related to this choice.

Below, we provide a summary of each filter and Instagram user who created it (the information was last updated on the 2nd of August 2022).

- Filter 0 is called `pretty` and was created by `heru sugiarta`,[16] a digital creator of filters with 300,000 followers on Instagram.
- Filter 1 is called `hari beauty` and was created by `hariani`,[17] a digital creator with 10.2 million followers on Instagram.
- Filter 2 is called `Just Baby` by `blondinochkavika`,[18] a creator of beauty filters with 179,000 followers on Instagram.
- Filter 3 is called `Shiny Foxy`, Filter 4 is called `Caramel Macchiato`, Filter 5 is called `Cute Baby Face`, all created by `sasha_soul_art`,[19] an Instagram filter designer of extremely popular filters on Instagram, with 1 million followers.
- Filter 6 is called `Baby cute face`, a popular beauty filter created by `anya__ilicheva`[20] with 13,500 followers on Instagram.
- Filter 7 is called `Big city life` by `triutra`,[21] a digital filter creator on Instagram with 138,000 followers.

All the users describe themselves as digital creators or digital filter creators and have created several Instagram filters, including the very popular beauty filters used in our study.

Note that the eight beauty filters selected through this approach reflect *feminine* beauty ideals. While it is impossible to quantitatively assess such a gender bias in the use of the filters, it is possible to grasp an intuition about it. To shed light on this topic, we performed search queries with relevant hashtags, such as #beautyfilter and related keywords on Instagram (both among posts and filters). In a qualitative and approximated manner, our searches revealed that the majority of users posting beautified content are women. We believe that gender biases related to beauty are, to some extent,

[16] https://www.instagram.com/herusugiarta/
[17] https://www.instagram.com/hariany/
[18] https://www.instagram.com/blondinochkavika/
[19] https://coveteur.com/sasha-soul-interview
[20] https://www.instagram.com/anya__ilicheva/
[21] https://www.instagram.com/triutra/

intrinsic in our society as a whole, and the popularity of beauty filters for women is one of its many manifestations.

Intended Use

OPENFILTER (section 3) is a flexible open framework to apply AR filters available in social media platforms on existing, publicly available large collections of images. We share this framework to provide the research community and practitioners with easier access to any AR filter available on social media, and to perform novel research in this emerging and culturally relevant field. We strongly discourage controversial and unethical uses of our framework and datasets. We acknowledge that, while the development of some applications could be appealing from a technical and scientific perspective, the subject matter of this work has a profound sociological and cultural component, which should not be ignored. As a consequence, we opt for protecting the general public from any consequence of this research, and thus share our datasets with exclusively a non-commercial license.

The intended uses of our datasets (FAIRBEAUTY and B-LFW) are very wide. Among the possibilities, we mention investigating the influence of beauty filters on social constructs, both computationally and through user studies. A second direction concerns societal implications of beauty filters. For example, these filters have raised concerns regarding existing biases in the automatic beautification practices and have been widely criticized for perpetuating racism and colorism. FAIRBEAUTY, in particular, opens the possibility of studying such issues computationally. As an insight, in Figure 6, we report exemplary images from FAIRBEAUTY, divided according to the label `race` in the original FAIRFACE dataset.

Figure 6: Examples of 70 different individuals in the FAIRBEAUTY dataset, divided by row according to the value of the label `race`.

Hosting and Maintenance plan

The project is version-trackable on our Github repository[22], where it will be permanently available. The datasets FAIRBEAUTY and B-LFW are hosted on Microsoft Azure[23], from where they can be downloaded.[24] The dataset was created at the ELLIS Unit Alicante, and the authors are committed to

[22] https://github.com/ellisalicante/OpenFilter

[23] We have requested official identifiers on http://identifiers.org/, respectively `fairbeauty` and `blfw`. The requests are currently being processed.

[24] Download links provided in the README files of the Github repository.

maintain the repository and the dataset storage at least until 2025, providing proper maintenance and development. Piera Riccio is in charge of supporting, hosting and maintaining the dataset. She can be contacted at her email address `piera@ellisalicante.org`.

For the time being, the authors do not foresee periodic updates of the dataset, but it will be corrected in case any error is detected in the current version. The availability of older versions will be subject to the type of update that is performed. The authors will make sure that any update is clearly communicated and justified to the rest of the community through the official GitHub page and the project page[25].

If other researchers are interested in collaborating on this work by extending or augmenting the datasets, they are warmly encouraged to get in touch with the authors. The authors will evaluate each proposal for extension before including it in the dataset. Even in this case, the authors will make sure to properly communicate the updates on both the GitHub and the project's page.

Licensing and Distribution

The datasets FAIRBEAUTY and B-LFW are distributed under the CC BY-NC-SA 4.0[26] license agreement, which allows sharing and re-adaptation for non-commercial purposes and redistribution under the same license.

The code for OPENFILTER is shared under a dual license. For non-commercial purposes, the GNU General Public License, version 2 applies.[27] Users interested in using the code for commercial purposes are asked to contact the authors for an explicit authorization. The authors will evaluate the ethical implications for each case.

Author Statement

We, the authors, will bear all responsibility in case of violation of rights.

[25] https://ellisalicante.org/datasets/OpenFilter
[26] https://creativecommons.org/licenses/by-nc-sa/4.0/
[27] https://www.gnu.org/licenses/old-licenses/gpl-2.0.html