ECE 492-45 Homework 11

Material Covered: PCA, *K*-Means, Deepfake

Problem 1 (20 points) [PCA] Complete ISLR 10.4.

Problem 2 (20 points) [PCA, *K*-Means] ISLR 10.6.1, 10.5.1.

Problem 3 (20 points) [PCA, *K*-Means] ISLR 10.7.10.

Problem 4 (20 points) [Deepfake] Refer to *Deepfake faceswap User Guide*.

Deepfake `faceswap` User Guide

Note 1: This user guide pieces together several writeups from the official GitHub repository for educational purpose. We do not claim any rights on this document. It is highly recommended to refer to the official guide for more details. The official guide may solve most problems you encounter.

Note 2: This package has multiple entry points. Before attempting to play with it, make sure you have read, understood and completed the installation instructions. If you are experiencing issues, first make sure you have installed the package correctly. Then you can refer to the workflow guide and the forum.

Step 1. Package Installation

Note: For macOS users, please refer to this link for installation.
If you have any questions, you can refer the official guide or the tutorial in the forum.

Method 1 – Installer (recommended): Windows and Linux now both have an installer which installs everything for you and creates a desktop shortcut to launch straight into the GUI. You can download the installer from this link.

Method 2 -- Manual Installation: You can follow the official guide step by step in the Manual Install section. The screenshot of the manual installation guide is shown in Fig. 1.

Fig. 1. Screenshot of the official manual installation section

Updating faceswap:

In the GUI, go to the Help menu and select "Check for Updates..." If updates are available, go to the Help menu and select "Update Faceswap." Restart Faceswap to complete the update.

Step 2. Workflow

If you have any questions, please refer to the official [workflow guide](#). It is highly recommended to work with the GUI. To swap faces, you will have to:

- Gather photos or videos.
- Extract faces from your raw photos.
- Train a model on the faces extracted from the photos/videos.
- Convert your sources with the model.

1) Photos/Videos Gathering

Here, we use swapping the faces of Trump and Nic Cage as an example. Possible sources for getting faces of Trump and Nic Cage are Google, DuckDuckGo and Bing image search. There are scripts to download large amounts of images. A better source of images are videos (from interviews, public speeches, or movies) as these will capture many **more natural poses and expressions**.

Once we have a folder full of pictures/videos of Trump and a separate folder of Nic Cage. Let's save them in our directory where we put the FaceSwap project. Example: `~/faceswap/src/trump` and `~/faceswap/src/cage`.

You may use your own face and a classmate's face for Deepfake.

Note 1: For better training results, you may need over 1000 images/frames for each person. The images should include as many poses, expressions, lighting conditions as possible.
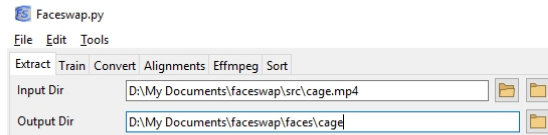
Note 2: FaceSwap can extract faces from both still images and video files. See Faces Extraction .

2) Faces Extraction

To extract trump from photos in a folder (Right hand folder icon):
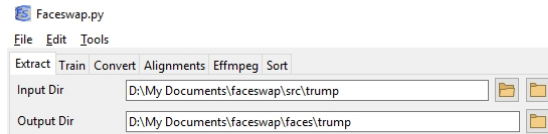
To extract cage from a video file (Left hand folder icon):

For input we either specify our photo directory or video file and for output we specify the folder where our extracted faces will be saved. The script will then try its best to recognize face landmarks, crop the images to a consistent size, and save the faces to the output folder. An `alignments.json` file will also be created and saved into your input folder. This file contains information about each of the faces that will be used by FaceSwap.

Note: this script will make grabbing test data much easier, but it is not perfect. It will (incorrectly) detect multiple faces in some photos and does not recognize if the face is the person whom we want to swap. Therefore: **Always check your training data before you start training**. The training data will influence how good your model will be at swapping.

If you have any questions, you can refer to the [tutorial](#) in the forum.

3) Model Training

Note 1: The training process will take a long time, which depends on many factors: the model used, the number of images, your GPU, etc. However, a ballpark figure is 12 to 72 hours on GPU and weeks if training on CPU.

Note 2: If you have any questions, you can refer to the [tutorial](#) in the forum.

Note 3: Different models and parameter setups might lead to different face swapping results. For this initial stage, it is recommended to use the Original model and follow the default setup.
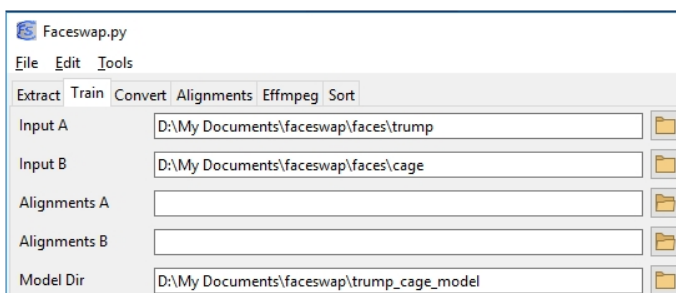
Once you run the command, it will start hammering the training data. If you have a preview up, then you will see a load of blotches appear. These are the faces it is learning. They don't look like much, but then your model hasn't learned anything yet. Over time these will more and more start to resemble trump and cage.

You want to leave your model learning until you are happy with the images in the preview. To stop training you can:

- Command Line: press "Enter" in the preview window or in the console
- GUI: Press the Terminate button

When stopping training, the model will save and the process will exit. This can take a little while, so be patient. The model will also save every 100 iterations or so.

You can stop and resume training at any time. Just point FaceSwap at the same folders and carry on.

4) Video Generation

Once we have trained our model, we can generate Deepfake videos as follows.

We generate `alignments.json` file for our swap. To do this, follow the steps in Faces Extraction: This time you want to run extraction for every face in your source video. This file tells the convert process where faces locate on source frames. You are likely need to clean up `alignments.json` by deleting false positives, badly aligned faces, etc. These will not result good on your final converted video. Just like extract you can convert from a series of images or from a video file.

Recall the initial example about Trump? Let's try to swap a face. We will use that directory as our input directory, create a new folder where the output will be saved, and specify which model to use.
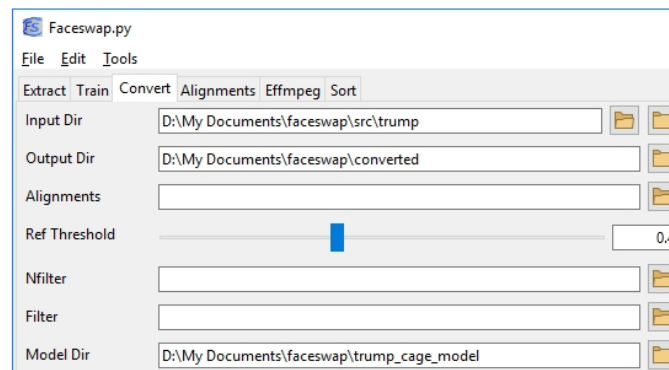
It should now start swapping faces of all these pictures.

Note: Sometimes, directly converting a video may not work. You may choose to covert the video to frames and then generate a video using the frames.