**EDIT 1/25**: A MEGA mirror of the app has been posted. The recent influx of downloads has caused Google Drive to temporarily take down the main download link.

**A tutorial has been created for using this app**[**here**](https://www.reddit.com/r/deepfakes/comments/7qih4e/fakeapp_v11_20180114_tutorial_in_laymans_terms/?st=jcgbkw8q&sh=3b1c1231)**.**

**EDIT 1/13**: Thanks for all the feedback and interest so far, I'm glad people have found the app useful. [Here is the download for the latest version, released yesterday (indirect, because direct DL links are filtered).](https://www.reddit.com/r/fakeapp/comments/7q5o7o/latest_version_download/?st=jcdmagwk&sh=9d4af47f)

I've completed a desktop app /w GUI to create deepfakes. [Here is a what it looks like.](https://imgur.com/BDX9ANb) For anyone unfamiliar with this subreddit, deepfakes are neural network-generated faceswap videos created with a machine learning algorithm designed by [/u/deepfakes](https://www.reddit.com/u/deepfakes). Check the sub wiki for more info. [Here is an excellent example of a deepfake of Daisy Ridley](https://thumbs.gfycat.com/EasySecondDouglasfirbarkbeetle-size_restricted.gif) produced with this app in less than a day by [/u/nuttynutter6969](https://www.reddit.com/u/nuttynutter6969). This app is intended to allow users to move through the full deepfake creation pipeline—creating training data, training a model, and creating fakes with that model—without the need to install Python and other dependencies or parse code. The download link is in the comments.

**Instructions:**

1. Download CUDA 8.0 and store it's bin folder in the [PATH environment variable](https://imgur.com/a/itUH9)
2. Split some videos with your two desired faces into two sets of a few hundred frames each with a tool like [FFMPEG](https://www.ffmpeg.org/). If you use FFMPEG, the command you want is: ffmpeg -i scene.mp4 -vf fps=[FPS OF VIDEO] "out%d.png". After splitting, run both directories of split frames through the "Extract" tool to produce training data
3. Switch to the "Train" tool, and input the paths of the training data produced in step 1 (it should be in a folder called "aligned") as well as the "models" folder along with this project (which you can move somewhere convenient)
4. Train until the preview window shows results you are satisfied with
5. Split the video to be faked into frames and run the "Convert" tool on them to create faked frames, which can then be remerged into a deepfaked video
6. Copy and reuse the same encoders for faster results in future fakes

**Requirements:**

-CUDA 8.0 must be installed, and its bin folder must be included in the PATH environment variable.

-At least a few GB of free space on disk to allow the app to create Temp files

**Notes:**

-Run fakeapp.bat to launch the app

-RuntimeError: module compiled against api version 0xc but this version numpy is 0xb is just a warning related to how the alignment libraries were installed, the app will run properly despite it appearing if no other errors occur

-It may take 30-45 seconds after pressing the Start button for the app to unpack and start the training/merging scripts the first time

-You can still quit training by focusing the training window and pressing "q"

-Paths to models/data must be absolute, not relative

**If it doesn't work for you:**

The console for the tool you are using (Align, Train, or Merge) will output a full error log if the tool fails. Here are some known errors with solutions:

*General Issues*

* All directories used by the app should have names comprised of only English characters. Many users have had issues with directories with Cyrillic or Chinese characters, so if you have directories like this make sure to use different directories with English character names when running the app.
* I Error message added in 1.1

*Extract Issues*

* If the Extract log contains from torch.\_C import \*, your computer is **out-date-on on Visual C++ Redistributable packages.** Download an update by Googling "vc redist." (I cannot include an actual link in this post.)
* Fixed in 1.1
* I Error message added in 1.1
* If the Extract log contains error while calling cudaMalloc() reason: out of memory, **you are probably training with images that are too large**. Make sure images are not greater than 1200x700 in resolution, this resolution is plenty to produce a good model

*Train Issues*

* If the Train log contains AssertionError in image\_augmentation.py **you are training with images of the wrong size**, make sure you are only training with the 256x256 images created using the Align tool
* If the Train log contains Missing cudart\_64.dll **you have the wrong version of CUDA installed** (Tensorflow requires 8.0)
* If the Train log contains MemoryError train.py line 60/line 61 **you are probably training with too many images at once for your GPU to handle** and should reduce the number of images you are training with (500 is more than sufficient)
* If the Train log contains OOM when allocating tensor with shape [W, X, Y, Z] the current model is too intensive for your GPU. **Try lowering the batch size (to a lower power of 2 than 64) and see if that helps.**
* I Fixed in 1.1

*Convert Issues*

* If Convert outputs faces with visible boxes around them, make sure that **Seamless is set to 'true' and faces are not too close up in the images you are working with**. The scripts this app works with cannot always avoid creating a box on very large faces, but will almost always create a seamless merge with moderately-sized faces.

In the future I plan to make ease-of-use improvements to the app and look into replacing scripts with more efficient/streamlined/accurate versions as they come out.