INFO7390: Advances in Data Science – Spring 2018

Face Swapping Technique with Auto Encoder

Sushrut Tadwalkar

[sushrutt.12@gmail.com](mailto:sushrutt.12@gmail.com)





<https://github.com/sushrutt12>

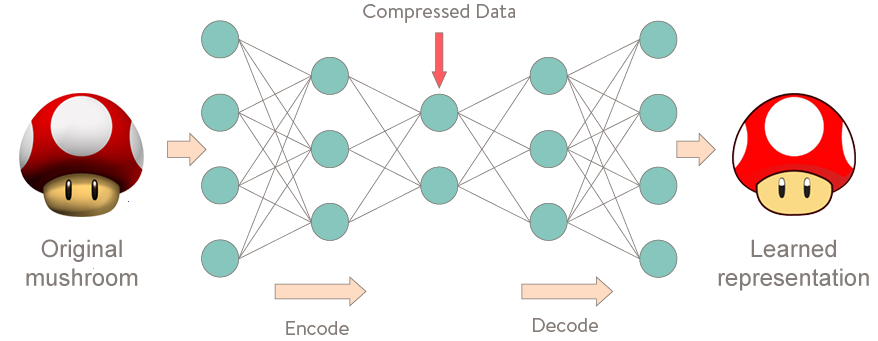
21st April 2018

**Abstract:**

## This paper attempts to elaborate the Deep Learning Technique used to replace a face of one person in an image with the face of another. This method uses CNN with Auto Encoder to perform this task (as seen on deepfakes [1]). We use two models (A & B), to Decode but a single Encoder. The main objective is to replace Robert Downey Jr.’s face with Chris Hemsworth’s face. For this task, we have gathered image data for each of the people. The model uses Mean Absolute Error as the loss function. The goal of our model is to generate clear images. We achieve this by reducing the error as much as possible. In the end, although the model learns the features of these two faces and we can swap their faces successfully, we don’t get the swapped images with 100% clarity.

**Introduction:**

A user on reddit, namely ‘Deepfakes’ published numerous fake videos of celebrities. These videos were adult in nature [1] and caused an uproar among the celebrities. Fake news was also on the rise thanks to this technique. So far, we have only seen this technique being misused for defaming people or spreading false information but it could be used for better purposes, such as in films this can be used to swap the face of a body double. Traditionally, if face swap was to be accomplished one could use any photo editing software to crop the face of one person and try to superimpose it over another. Assuming it takes 5 minutes to work on one image, consider a video of 1 minute with 24 fps (frames per second range varies from 23-30), this would make 1440 images (60 seconds x 24 fps) and the time spent to replace the face for these would be 5 days (1440 images x 5 minutes=7200 minutes / 60 mins = 120 hours / 24 hours = 5 days) not including room for human error. Using faceswap technique production houses could lower the production cost of movies significantly. The deep learning model used for this technique is CNNs with Auto Encoders. By far CNN’s have been assumed to work best with images [2] as machine learning models in the image processing domain. CNN’s are powerful because they can extract features which would be otherwise difficult for the human eye to decipher and extract manually.

*The following picture depicts how an auto encoder works [3]:* *The image is fed to the input layer, this is where the encoding starts. After a few layers of reducing the neurons in each layer the features are stored in a compressed form. The network is then forced to express these features in increasing order of neurons per layer, this process is decoding of the features and results in a learned representation of the original image.*

Using Deep Learning the model extracts features from the image that it deems important. A sparse Auto Encoder sits as the most vital part of this technique. Sparse Auto Encoders have been known for optimizing the image resolution. We use it to make our Encoder learn what a face looks like and try to reconstruct the same face.

**Dataset Details:**

There are two datasets for training our model, one is Chris Hemsworth’s images & the second is Robert Downey Jr’s images. We have obtained close to 2100 total images from GettyImages [4] the dimensions of these images lie in the range 408 x 612 -1024 x 684. These images also contain other people in them and therefore we will have to extract the faces of the ones we require.

**Methods:**

**I. Extraction of Faces:**

Using OpenCV we extract the faces from the images. The number of face images are in the range 350 – 450. Each image is of dimension 256 x 256 and have been picked based on clarity.

*The following are examples of the original image, followed by images where the face is extracted (Top: Chris Hemsworth, Bottom: Robert Downey Jr.)*

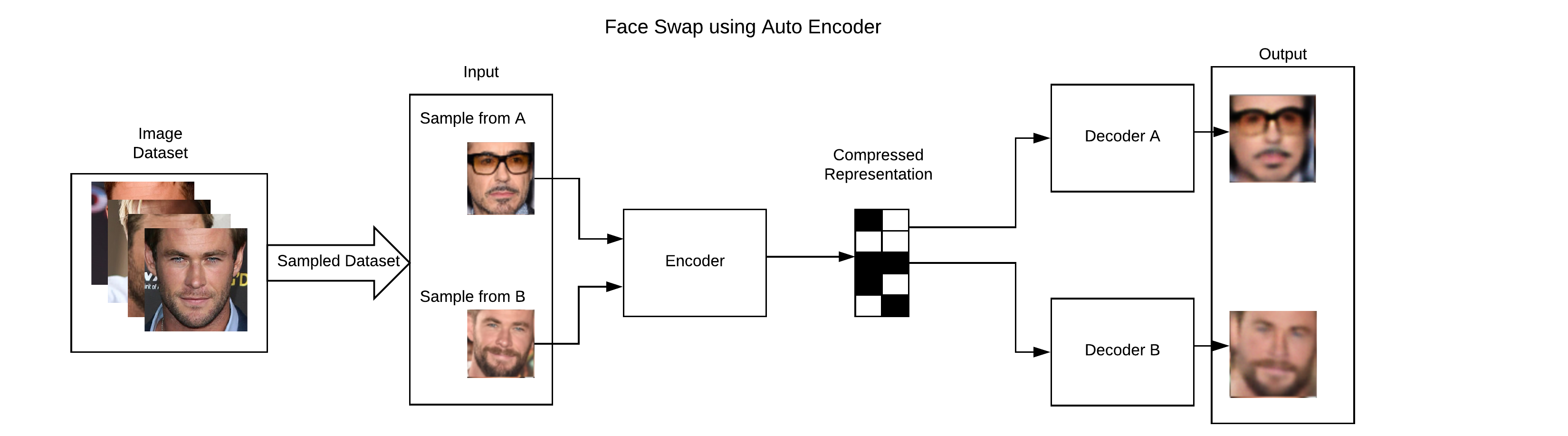




**II. Algorithm & Training:**

We will be using a single Encoder and two Decoders to train our model. For the Encoder we use 4 Convolutional layers with activation LeakyReLU and neurons double at each layer, followed by Dense and Flatten. For the Decoder we use 3 Convolutional layers with activation LeakyReLU, followed by another Convolutional layer with Sigmoid activation. We use Adam as the optimizer with learning rate 0.00005 . The loss function used is Mean Absolute Error. More details can be found on Github[5]. This is based on the post found on GithubGist [6], posted by a reddit user[7].

*The following image depicts the working of our model:*



*The following image depicts the training progress of our model. This progress is divided into groups of 3 columns. The first column represents the input image, the second column represents the output image, the third column represents the face swapped image.*



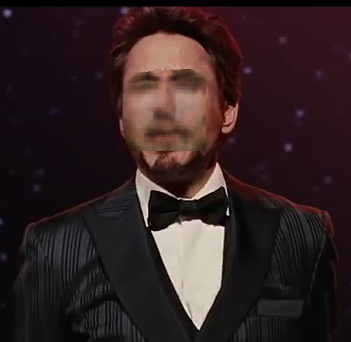
We have trained this model for close to 1500 epochs to obtain Loss for Model A: 0.04723, Loss for Model B: 0.06410. But for best results (clearer image) it is recommended to train till the loss drops to about 0.01

**III. Conversion of frames**

After finding a video we convert that video into frames using ffmpeg [8]. We use our model to identify the face and swap it. OpenCV creates a mask and then the face of the target is superimposed on the image. After this process is completed for each frame we stitch the video back together.

**Result**

*The following image depicts the result of the face replacement:*



Since the error rate was still 0.04723, we couldn’t obtain significant clarity.

|  |  |  |  |
| --- | --- | --- | --- |
| Error for A | Error for B | Batch Size | Epoch |
| 0.05862 | 0.07512 | 64 | 1000 |
| 0.05322 | 0.07353 | 64 | 1500 |
| 0.05226 | 0.07171 | 128 | 1000 |
| 0.04723 | 0.06410 | 128 | 1500 |

The video can be found on YouTube [9]

But a few thoughts to improve this model would be to train the model for more epochs, get higher quality dataset, try various techniques such as GAN’s.

**References**

[1] <https://en.wikipedia.org/wiki/Deepfake>

[2] <http://cs231n.github.io/convolutional-networks/>

[3] <http://curiousily.com/data-science/2017/02/02/what-to-do-when-data-is-missing-part-2.html>

[4] <https://www.gettyimages.com/>

[5] <https://github.com/sushrutt12/faceswappingtechnique>

[6] <https://gist.github.com/anonymous/d3815aba83a8f79779451262599b0955>

[7] <https://www.reddit.com/r/deepfakes/>

[8] <https://www.ffmpeg.org/>

[9] <https://youtu.be/FZXQtbHG6-o>

<https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/>

<https://en.wikipedia.org/wiki/Autoencoder>

<https://www.deepfakes.club/faq/>

<https://github.com/deepfakes>