# BBM467 - Blog Post Project

Team Octopus

21946309, Ahmet Karaca
21946113, Oğuzhan Ertekin

## Deep Fake Technology

### What is Deepfake? How does it operate?

Deepfake is a form of media in which a picture of one person is replaced with another using artificial neural networks. The name "Deepfake" is derived from the underlying artificial intelligence (AI) technique known as "deep learning." To create phony media that seems genuine, deep learning algorithms are utilized to swap faces in videos and digital material. These algorithms train themselves to solve issues when given enormous amounts of data.

Although there are various ways to produce Deepfake media, the most popular one makes use of face-swapping autoencoders in deep neural networks using autoencoders. To serve as the foundation for the Deepfake, a series of video clips of the person you want to put in the target must come first, followed by a target video. The target video may be a clip from a Hollywood film, for instance, while the films of the person you want to place in the movie could be unrelated footage you acquired from YouTube.

A deep learning AI algorithm called the autoencoder is tasked with watching the video clips to learn how the person appears from various perspectives and in various environments and then mapping that person onto the person in the target video by identifying shared traits.

Generative Adversarial Networks (GANs), another kind of machine learning, are incorporated into the process. GANs identify and fix any Deepfake problems over the course of several rounds, making it more challenging for Deepfake detectors to identify them.

In order to "learn" how to create fresh instances that closely resemble the actual thing, GANs are also frequently utilized as a popular technique for the production of DeepFake.

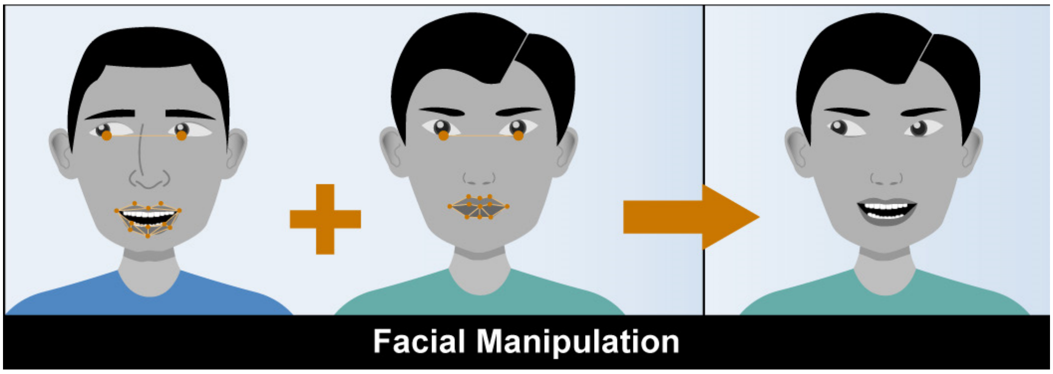

Figure 1. Face Manipulation

*Source:* [*https://www.gao.gov/products/gao-20-379sp*](https://www.gao.gov/products/gao-20-379sp)

## Advantages of Deepfake

- Deepfake may be utilized as a creative medium to revive historical figures. As an illustration, a painting of the Mona Lisa may be utilized to create a computer-generated image of a talking Mona Lisa.
- AI avatars may be created using Deepfake technology and utilized in training videos. Since lockdowns and health concerns have made video shootings with actual humans much more challenging to pull off, startups like London-based Synthesia have been drawing greater attention from the business world amid the COVID epidemic.
- In order to test clothing or new haircuts before trying them on in person, Deepfake may be utilized to build individual avatars.
- Deepfake can also be used for identity protection and anonymization in various fields like investigative news reporting, finance, etc.

## Disadvantages of Deepfake

- With the use of modified celebrity videos, Deepfake may be utilized to distribute false information.
- Deepfake may also be used to launch deceptive social media campaigns that influence public opinion and have unfavorable effects.
- Politicians have also employed Deepfake videos. For instance, a Belgian political party broadcast a video of Donald Trump speaking and urging Belgium to leave the Paris climate accord in 2018. But the address was a deep fake; Trump never gave it. Deepfake has previously been used to produce deceptive films, and tech-savvy political gurus are preparing for a new wave of fake news that will incorporate this impressively realistic Deepfake.

## Does Deepfake only consist of videos?

Deepfake isn’t merely found in videos. A rapidly expanding field with a vast array of uses is Deepfake audio.

With just a few hours (or, in some cases, minutes) of audio of the person whose voice is being cloned, a realistic audio Deepfake can now be created using deep learning algorithms. Once a model of a voice is created, that person can be made to say anything, as was the case last year when the fake audio of a CEO was used to commit fraud.

## How to recognize Deepfake?

As Deepfake spreads, society as a whole will likely need to acclimate to seeing Deepfake films in the same way that internet users have grown skilled at spotting other types of fake news.

In order to identify and stop Deepfake technology from spreading, which might start a vicious cycle and perhaps do more harm, it frequently takes more advanced Deepfake technology to be developed, as is the case with cybersecurity.

A few signs may be used to identify Deepfake:

- Videos where the person never blinks, blinks too frequently, or blinks in an unnatural way are the consequence of the current deepfake's struggles to convincingly animate faces. But a fresh Deepfake that didn't have this issue was issued when researchers from the University of Albany published a report identifying the irregular blinking.
- Keep an eye out for facial issues, skin or hair issues, and faces that look blurrier than the surroundings in which they are situated. the possible abnormally soft appearance of the focus.
- Is the lighting unnatural-looking? The lighting of the clips that served as models for the false video is frequently kept by Deepfake algorithms, even when it is not a good match for the lighting in the target video.
- If the video was fabricated but the original audio was not carefully modified, the audio could not seem to fit the person.

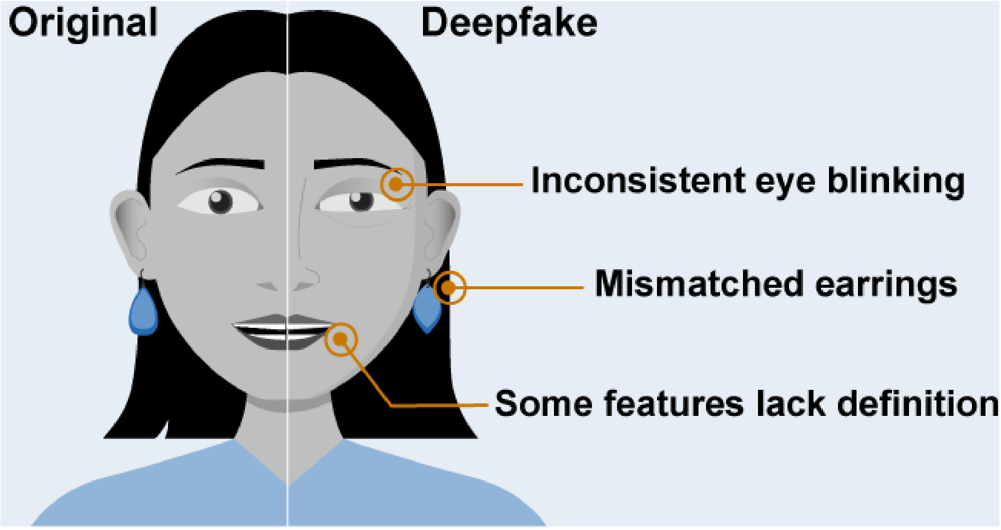

Figure 1. Points to consider in deepfake detection

*Source: https://www.gao.gov/products/gao-20-379sp*

### How to create a simple Deepfake video using Python?

*1- Install "PyYAML" and "imageio-ffmpeg" with terminal.*

```
!pip install PyYAML==5.3.1

!pip install imageio-ffmpeg
```

*2- Import the necessary packages to be used.*

```
import imageio
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from skimage.transform import resize
from IPython.display import HTML
import warnings
from demo import load_checkpoints
from demo import make_animation
from skimage import img_as_ubyte
```

*3- "warnings.filterwarnings("ignore")", is a Python statement that tells the interpreter to ignore any warning messages that are generated by the script. The next two lines of code use the "load_checkpoints" function to load two objects: a generator and a keypoint detector. The function takes two arguments: the path to a configuration file and the path to a checkpoint file.*

```
warnings.filterwarnings("ignore")
generator, kp_detector = load_checkpoints(config_path='config/vox-256.yaml',
                            checkpoint_path='.../vox-cpk.pth.tar')
```

*4- The following two lines of code use the "imageio.imread" function to read a source image and a driving video from a disk. The source image is then resized to a 256x256 resolution using the resize function from the scikit-image library. The driving video is also resized to a 256x256 resolution, but the resizing is applied to each frame of the video.*

```
source_image = imageio.imread('../test_picture.png')
driving_video = imageio.mimread('../test_video.mp4',memtest=False)

#Resize image and video to 256x256
source_image = resize(source_image, (256, 256))[..., :3]
driving_video = [resize(frame, (256, 256))[..., :3] for frame in
driving_video]
```

*5- The display function takes three arguments: a source image, a driving video, and a generated video. It creates a Matplotlib figure and animates the three videos by concatenating them horizontally and displaying them as a sequence of frames. The function returns an animation object, which can be converted to an HTML5 video using the to_html5_video method.*

```
def display(source, driving, generated=None):
    fig = plt.figure(figsize=(8 + 4 * (generated is not None), 6))

    ims = []
    for i in range(len(driving)):
        cols = [source]
        cols.append(driving[i])
        if generated is not None:
            cols.append(generated[i])
        im = plt.imshow(np.concatenate(cols, axis=1), animated=True)
        plt.axis('off')
        ims.append([im])

    ani = animation.ArtistAnimation(fig, ims, interval=50,
repeat_delay=1000)
    plt.close()
    return ani
```

*6- Finally, the make_animation function is called with the source image, driving video, generator, and keypoint detector as arguments. This function generates the animation by applying the motion model to the input data. The resulting animation is then saved to a file and displayed in the output.*

```
predictions = make_animation(source_image, driving_video, generator,
kp_detector, relative=True)

#save resulting video
imageio.mimsave('../generated.mp4', [img_as_ubyte(frame) for frame in
predictions])
#video can be downloaded from /content folder

HTML(display(source_image, driving_video, predictions).to_html5_video())
```

**Sample Outputs:**

**[https://drive.google.com/drive/folders/1ejIeB2JnLbTgp8kViy32L3rJAkC5umZ9?usp=sharing](https://drive.google.com/drive/folders/1ejIeB2JnLbTgp8kViy32L3rJAkC5umZ9?usp=sharing)**

For a more detailed tutorial, you can access the YouTube video we referenced from the link below.
[Make DeepFake Video Step by Step Using Python | KNOWLEDGE DOCTOR | Mishu Dhar]()

**References**

1.https://www.theguardian.com/technology/2020/jan/13/what-are-deepfakes-and-how-can-you-spot-them
2.https://www.businessinsider.com/guides/tech/what-is-deepfake
3.https://balavivek.medium.com/deepfake-with-python-5195b320f267
4.https://www.analyticsvidhya.com/blog/2021/10/an-introduction-to-deepfakes-with-only-one-source-video/
5.https://youtu.be/bHd9xRD3Trw?list=LL