# Lesson 1.1: Getting the Data

**Objective**

* Download [DeepfakeTIMIT database](https://liveproject-resources.s3.amazonaws.com/other/detectingdeepfakes/DeepfakeTIMIT.tar.gz) to obtain the set of Deepfake videos. Once the database is downloaded, read the Readme file and playback several of the Deepfake videos (there are high- and low-quality Deepfakes) to see what they look like and how they are different.
* Download the original [VidTIMIT dataset](https://liveproject-resources.s3.amazonaws.com/other/detectingdeepfakes/VidTIMIT.zip), and check the included videos.

**Workflow**

1. Go to the download link at [DeepfakeTIMIT database page](https://liveproject-resources.s3.amazonaws.com/other/detectingdeepfakes/DeepfakeTIMIT.tar.gz).
2. Download the dataset, extract the data from the archive, and read the Readme file inside. Then, browse and playback some videos in the dataset to see the Deepfake versions.
3. Go to [VidTIMIT original videos](https://liveproject-resources.s3.amazonaws.com/other/detectingdeepfakes/VidTIMIT.zip) and download the archives listed there. Extract the data from the archive. Browse and playback some of the original videos.

# Lesson 1.2: Processing the Videos

**Objective**

* Use [OpenCV](https://opencv.org/) to read videos frame by frame. Use matplotlib or OpenCV to save the frames as images. Use OpenCV to compare frames from genuine and Deepfake videos. You can compute a simple difference between original and Deepfake frames and save the difference as an image. You can also compare the histograms of original and Deepfake frames.

**Workflow**

1. Use Python Glob library to recursively iterate through both real and fake videos folders.
2. Use a popular OpenCV library for image and video processing and analysis to loop through the frames of each video.
3. Save frames as JPEG images for one original (your choice) and the corresponding Deepfake video to the disk. You can use matplotlib or OpenCV to do so.
4. Take one frame from an original video and the same frame from the corresponding Deepfake. Find difference between these two frames and save it as an image to visually see the difference.
5. Compute histograms for these two frames (one from original video and the corresponding one from Deepfake) and visualize them with matplotlib. What does the difference look like?
6. Observe what makes video frames from Deepfake videos look different compared to originals.