# YouTube Whisper Colab Notebook Workflow Analysis

This document outlines the workflow, dependencies, and user interactions based on the youtube\_whisper.ipynb Colab notebook.

**Objective:** Transcribe audio from a YouTube video using OpenAI’s Whisper model and provide the transcript in text (.txt) and/or subtitle (.srt) formats.

**Platform:** Google Colaboratory (requires GPU runtime, T4 recommended).

**Core Dependencies:**

* pytube: Python library used to download video streams (specifically audio) from YouTube.
* openai-whisper: OpenAI’s library for automatic speech recognition (ASR). Installed directly from GitHub (git+https://github.com/openai/whisper.git).
* torch: PyTorch library, a dependency for Whisper.
* Standard Python Libraries: os, re, pathlib.

**User Inputs:**

1. **YouTube URL:** The web address of the video to be transcribed.
2. **Whisper Model:** Selection from available Whisper model sizes (tiny, base, small, medium, large, large-v2, large-v3). Larger models are generally more accurate but require more computational resources (VRAM) and take longer.
3. **Output Formats:** Boolean flags to select whether to generate a plain text file (.txt) and/or an SRT subtitle file (.srt).

**Workflow Steps (as per notebook cells):**

1. **Configuration (Cell 1):**
   * User manually enters the YouTube URL into a variable (YouTube\_URL).
   * User selects the desired Whisper model size (whisper\_model).
   * User sets boolean flags (text, srt) for desired output file types.
2. **Dependency Installation (Cell 2):**
   * Installs pytube and openai-whisper using pip.
   * Imports necessary Python libraries (os, re, torch, pathlib, pytube, whisper, whisper.utils.get\_writer).
   * This step typically only needs to be run once per Colab session.
3. **Execution (Cell 3):**
   * **Device Setup:** Detects if a CUDA-enabled GPU is available and sets the device accordingly (cuda or cpu).
   * **Model Loading:** Loads the selected Whisper model (whisper\_model) onto the chosen device.
   * **Audio Download:**
     + Uses the pytube library to connect to the provided YouTube\_URL.
     + Filters available streams to find the highest quality audio-only MP4 stream.
     + Downloads this audio stream to the Colab environment. The filename is automatically generated based on a sanitized version of the video title (using the to\_snake\_case helper function).
   * **Transcription:**
     + Calls the model.transcribe() method, passing the path to the downloaded audio file.
     + The verbose=False argument suppresses detailed transcription progress output.
   * **Output File Generation:**
     + If the text flag is True, opens a .txt file (named after the video title) and writes the transcribed text content (result["text"]) into it.
     + If the srt flag is True, uses Whisper’s built-in get\_writer("srt", ...) utility to generate an SRT formatted subtitle file.
   * **File Download (Colab Specific):**
     + Uses google.colab.files.download() to trigger a browser download prompt for any generated .txt or .srt files residing in the Colab environment’s /content/ directory.

**Key Processes:**

* Fetching and downloading audio content from YouTube.
* Running a sophisticated AI model (Whisper) for speech-to-text conversion.
* Formatting the output into standard text and subtitle files.
* Interacting with the Colab environment for file system operations and triggering downloads.