
Video Translation Server

Saturday, 10.26.2024

Overview

This project consists of a Flask server that simulates the status of video translation jobs and a client library to interact with the server.

Installation

Make sure to install the required packages: Flask, pytest.

Command: `pip install Flask requests pytest`

Running the Server

Command to run the server: `python server.py`

Running the Tests

You can run the tests using the following command: `pytest test_integration.py`

Client Library Documentation

Overview

The Video Translation Client Library provides an interface to interact with the Video Translation Server. This library allows users to start translation jobs and check their status through a simple API.

Installation

Ensure you have the required packages installed. If you haven't done this already, you can install the necessary dependencies using pip: `pip install requests`

Usage

Step 1: Importing the Library

First, import the `VideoTranslationClient` class from the client library:

```
from video_translation_client import VideoTranslationClient
```

Step 2: Creating an Instance of the Client

To create an instance of the `VideoTranslationClient` by providing the base URL of the server: `client = VideoTranslationClient("http://localhost:8000")`

Step 3: Check Status of Translation Jobs

You can check the status of the translation job using the `check_status` method. This method will poll the server for the current job status, printing the results to the console.

```
client.check_status(interval=2, timeout=30)
```

Step 4: Example

Here's a complete example demonstrating how to use the client library:

```
from video_translation_client import VideoTranslationClient
```

```
client = VideoTranslationClient("http://localhost:8000")
```



```
client.check_status()
```

Features

- **Status Checking:** The library provides functionality to retrieve the current status of the video translation job.
- **Configurable Polling:** You can specify how frequently to check for status updates and how long to wait before timing out.

Error Handling

The client library includes basic error handling for network issues and unexpected responses. If an error occurs while fetching the status, an exception will be raised, and a message will be printed.