CLIP Fine-Tuner FastAPI

1 Introduction

This project provides a FastAPI application to serve a fine-tuned CLIP model for image classification. Users can upload an image and specify a model name to receive predictions based on predefined subcategories.

2 Prerequisites

Ensure you have the following installed:

- Python 3.8 or higher

- FastAPI

- torch (PyTorch)

- clip (CLIP)

- PIL (Pillow)

- requests

You can install the required packages using pip:

```
Installation Command
pip install fastapi torch pillow requests
```

3 File Structure

- `app.py`: FastAPI application for serving the fine-tuned CLIP model.

- `predict.py`: Command-line utility to interact with the FastAPI endpoint.

4 Setup

4.1 Model Files

Ensure that you have your fine-tuned CLIP model files in the `models/clip/` directory. The model file paths should be specified in the `model_info` dictionary within `app.py`.

4.2 Run the FastAPI Server

Start the FastAPI server by running:

```
─────────── Server Start Command ────────────
uvicorn app:app --reload
```

The server will be available at `http://127.0.0.1:8000`.

5 Usage

5.1 API Endpoint

The FastAPI application exposes the `/predict/` endpoint for image classification. To get a prediction:

- **Method:** POST

- **URL:** `http://127.0.0.1:8000/predict/`

- **Parameters:**

  - `file`: The image file to be classified.
  - `model_name`: The name of the model to be used for classification. (Must match one of the keys in `model_info`)

5.2 Example Request

You can test the API using the provided `predict.py` script. Run the script with the following command:

```
─────────────── Example Command ────────────────
python predict.py <image_path> <model_name>
```

Replace `<image_path>` with the path to your image file and `<model_name>` with one of the predefined model names.

Example:

```
──────────────── Example Usage ─────────────────
python predict.py my_image.jpg hineng
```

The script will print the JSON response from the server, which includes the predicted class.

6 Error Handling

If an invalid model name is provided or if an error occurs during processing, the API will return an error message.