Project Title: Imagedata_extraction

About the project:

This project mainly focuses on the inaccuracies and inefficiencies of manually extracting data from bilingual invoices. Existing solutions are insufficient for international enterprises since they are sometimes restricted to particular languages or formats. It is difficult to create a reliable system that can process a wide range of invoice formats and languages, precisely extracting the required data and storing it in a uniform format.

Prerequisites:

- Python
- Git
- OCR
- Computer Vision

What will you learn?:

In this project we will learn about how we can implement as well as integrate computer vision with Natural language processing with the help of OCR(optical character recognition), in this project, we will create a DL model that will decrypt the information present within an image

Skills you will practice:

- Python
- ML
- DL
- AI
- Gen AI

How to execute? Your learning platform:

- you can refer to either Google Collab or lightning.ai platform for the implementation of the codebase of the project
- Install all the necessary libraries that are already present in the requirements.txt file by running the following command: pip install -r requirements.txt
- you can implement the codebase onto localhost desktop or cloud.

Learn step-by-step:

Hour 1: Setting Up GitHub and Git

1. Create a GitHub Account (10 minutes):

- Go to github.com and sign up for a free account.
- Choose a username and password.
- Verify your email address.

2. Install Git (15 minutes):

- For Windows:
  - Download the latest Git installer from git-scm.com.
  - Follow the on-screen instructions during installation.
- For macOS:
  - Use Homebrew to install Git: `brew install git`
- For Linux:
  - Use your package manager (e.g., `sudo apt install git` on Ubuntu/Debian, `sudo yum install git` on Fedora/CentOS)

3. Configure Git (10 minutes):

- Open your terminal or command prompt.

- Set your user name and email:

  Bash

  git config --global user.name "Your Name"
- git config --global user.email "your_email@example.com"

- Check your configuration:

  Bash

  git config --list

4. Create a New Repository on GitHub (5 minutes):

- Log in to your GitHub account.
- Click the "New repository" button.
- Give your repository a name and description.
- Choose the repository visibility (public or private).
- Click "Create repository."

5. Clone the Repository Locally (10 minutes):

- Copy the HTTPS or SSH URL of your repository from GitHub.
- Open your terminal and navigate to your desired project directory.
- Clone the repository:

  Bash

  https://github.com/Mudrikkaushik/Imagedata_extraction.git

Hour 2: Installing Dependencies with `requirements.txt`

1. Understand `requirements.txt` (5 minutes):

- Explain the purpose of `requirements.txt`.
- Show how it lists packages and their versions.

2. Create a Virtual Environment (10 minutes):

- Install `virtualenv`:

  pip install virtualenv

- Create a virtual environment:

  virtualenv venv

- Activate the virtual environment:
  - Windows: `venv\Scripts\activate`
  - macOS/Linux: `source venv/bin/activate`
-

3. Install Dependencies (10 minutes):

- Install all packages listed in `requirements.txt`:

  Bash

  pip install -r requirements.txt

4. Basic Git Workflow (25 minutes):

- Committing Changes:
  - Stage changes: `git add .`
  - Commit changes: `git commit -m "Your commit message"`
- Pushing Changes to GitHub:
  - Push to the remote repository: `git push origin main`
- Pulling Changes from GitHub:
  - Fetch changes: `git fetch origin`
  - Merge changes: `git merge origin/main`
- Resolving Conflicts:

    - Briefly explain how to resolve merge conflicts using Git's merge tools.

Additional Tips:

- Encourage the use of clear and concise commit messages.
- Explain the importance of regular commits.
- Demonstrate how to use Git branches for feature development and bug fixes.
- Introduce the concept of a `.gitignore` file to exclude unwanted files from version control.
- Provide resources for further learning, such as Git tutorials and documentation.

Course Objectives:

> This project aims to create a multilingual invoice extractor that can automatically extract important data from invoices in various formats and languages. A CSV file contains the retrieved data for convenient access and analysis. The main goals are to enhance data extraction accuracy, and multilingual support, and ensure efficiency and scalability. The creation of a fundamental extraction module, integration with language translation providers, and preliminary testing using sample invoices are all part of the midterm work**.**

Project Goals:

1. Create a system that can extract invoice data from many languages: Construct a scalable and adaptable system that can process invoices in a variety of formats and languages.

2. Make sure the data extraction process is highly accurate and efficient. Increase the precision and effectiveness of data extraction by utilizing cutting-edge technologies like Natural Language Processing (NLP) and Optical Character Recognition (OCR).

3. Save the data that has been extracted in a CSV file. Make sure the retrieved data is saved in an easily accessible, standard format for analysis.

4. Combining language translation services: Integrate your system with translation services to enable data extraction from invoices in several languages.

Course Structure:

The project employs a methodical approach with multiple crucial steps:

1. **Literature Review**: Perform a thorough analysis of the technology and procedures currently in use for multilingual OCR and invoice data extraction.

2. **Requirement analysis:** Determine and record the necessary languages, formats, and essential data points to be extracted for the multilingual invoice extractor.

3. **System Design:** Create the system architecture, choosing the technologies and tools for data storage, OCR, and natural language processing.

4. **Implementation:** Create the software solution by combining different parts, including data storage modules, translation services, and OCR engines.

5. **Testing and Validation:** To verify correctness and efficiency, test the system using a range of invoices in various languages and formats. Verify the outcomes with data that was manually extracted.

6. **Maintenance and System Deployment:** Install the system and provide regular upgrades and maintenance to handle new formats, languages, and growing issues.

Meet your educator:

Hi, my name is Mudrik Kaushik, a Corporate Trainer and automation Trainer manifesting to make a difference in the field of Artificial Intelligence, Data Science,

Machine learning, and cyber security. moreover, I am a human being who's trying to live a different

References:

1. Gelb, A. Applied Optimal Estimation. Cambridge, MA: M.I.T. Press, 1974.

2. Kalman, R.E., & Pucy, N.S. "New results in linear filtering and prediction theory." Trans. ASME, J. Basic Eng., Vol. 83-D, pp. 95-108, Mar. 1961.

3. Vidyasagar, M., & Bose, N.K. "Input-output stability of linear systems defined over measure spaces." In Proc. Midwest Symp. Ciro. Syst., Montreal, P.O. Canada, Aug. 1975, pp. 394-397.

4. Viera, A.C.G. "Matrix orthogonal polynomials, with applications to autoregressive modeling and ladder forms." Ph.D. Dissertation, Stanford Univ., Stanford, CA, Dec. 1977.

5. Wonham, W.M. (1982) Private Communication.

6. Tesseract OCR Documentation, https://github.com/tesseract-ocr/tesseract

7. Google Cloud Vision API Documentation, https://cloud.google.com/vision/docs

8. ISO/IEC 27001:2013. Information technology — Security techniques — Information security management systems — Requirements.

9. NLTK Documentation, https://www.nltk.org

10. Spacy Documentation, https://spacy.io