**Automated Invoice Processing System**

**Key Components:**

1. Identifying metadata: Choosing appropriate data to extract is challenging if the source system is poorly documented.
2. Image Preprocessing
3. Implementation of OCR technique
4. Data Extraction using NLP
5. Create database to store the extracted data

**Image Preprocessing Techniques:**

Image preprocessing techniques are applied to increase the quality of the images. This include adjusting the brightness and contrast, removing noise, and correcting any rotated images. These adjustments improve the accuracy of the OCR system.

* **Image Geometric Transformations**
* **Page Rotation**: The orientation of image should be corrected before text recognition.
* **Deskew or Skew Correction**: Most Images captured from flatbed scanners or photographed by digital cameras are slightly skewed. So, Detecting the skew angle and rotating the page so that text will appear horizontal and not tilted in any angle.
* **Lines Straightening:**When the lines are curvy as in the case of the above image, it may result in OCR issues and can cause issues with line segmentation and text re-arrangement. Hence, detecting the curved lines and straightening them will improve our OCR results.

**Noise Removal Techniques:**

Blurring or Smoothing of an image removes “outlier” pixels that may be noise in the image. There are many filters that can be used to blur images and each has its own advantages and disadvantages.

1. **Gaussian Blur:** Uses Gaussian kernel for convolution and good at removing Gaussian noise from the image. It is much faster compared to other Blurring techniques but fails to preserve edges which may affect OCR output.
2. **Median Blur:** Replaces the central element in the kernel area with the median value of the pixels under the kernel area. It is good at removing salt and pepper noises from the scanned document.
3. **Bilateral Filtering:** It is highly effective in noise removal while keeping the edges sharp. Along with the Gaussian filter in space, it also takes another Gaussian filter which is a function of pixel difference. The Gaussian function of space makes sure that only nearby pixels are considered for blurring, while the Gaussian function of intensity difference makes sure that only those pixels with similar intensities to the central pixel are considered for blurring. So it preserves the edges since pixels at edges will have large intensity variation.

**Implementation of OCR technique**

Below are the some of the OCR techniques used:

1. Pytesseract
2. EasyOCR
3. [Doctr](https://www.analyticsvidhya.com/blog/2024/04/ocr-libraries-in-python/#h-doctr)
4. [GOCR](https://www.analyticsvidhya.com/blog/2024/04/ocr-libraries-in-python/#h-gocr)
5. Keras-OCR
6. [Amazon Textract](https://www.analyticsvidhya.com/blog/2024/04/ocr-libraries-in-python/#h-amazon-textract)

Modules used are:

* cv2
* numpy
* easyocr
* matplotlib

Example method of extracting data from input image and extracting data from image using easyOCR with code example is checked in to github also it is attached below:



The purpose of NER is to automatically extract the structed information from unstructured text, enabling machines to understand and categorize entities in a meaning manner for various applications like text summarization, building knowledge graphs, question answering and knowledge graph construction.

Methods of NER in NLP include:

* Lexicon based NER.
* Rules Based
* ML Based
* Deep learning Based.

**Text Classification using Named Entity Recognition**

Named Entity Recognition ([NER](https://spacy.io/usage/training/#ner)) is one of the text classification techniques used in Natural Language Processing. It is also known as entity identification, entity chunking and entity extraction. It is designed to locate and classify named entity described in unstructured data into pre-defined categories such as name, country, organization, currency etc. NLTK and [SpaCy](https://spacy.io/) are the most used python packages for NER. We can build the NER model using SpaCy. We can take the output of the OCR and train a custom NER model in order to classify the following entities.

1. Invoice Number
2. Invoice Date
3. Currency
4. Total Amount
5. Purchase Order

**Steps to perform training NER Model**

1. Create dataset
2. Training NER model
3. NER Model Evaluation
4. Testing and Validation
5. Deployment for inference

After training we will store the model into a predefined directory. The trained model classify the input text extracted from inference and classify afore mentioned entities along with tokenization. [Displacy](https://spacy.io/usage/visualizers) component has been used to visualize the classification output. The extracted text can be stored into a dictionary and the same can be used to store into a CSV, SQL or directly push to SAP system.

**End to End pipeline for deployment using MLOps**

The model can be deployed in AWS. A pipeline tool designed to transmit the image to AWS via WIFI or ethernet to Amazon Kinesis Video Streams

* KVS enables to stream live image and other time-encoded data from devices like cameras to the AWS cloud.
* We can combine KVS and other AWS services like Amazon Rekognition (for image and video analysis)
* Model is deployed in AWS EC2 instance by creating an end point.
* Images will be stored in S3 bucket for review
* SQL can be integrated to AWS instance to store the data.

Tools required:

1. Python
2. Front end Flask application
3. Dockers
4. AWS S3 Bucket
5. AWS EC2 Instance
6. Git hub to creating CI/CD pipeline