Follow up Question Answers

If you can only leverage libraries and models which you can deploy in a private cloud environment, which ones are you likely to want to explore/experiment with further?

For text extraction from images in a private cloud environment, you would likely want to explore libraries and models that can be deployed privately. Then all the below listed packages or API would be some of the options:

- Tesseract (via pytesseract)
- EasyOCR
- Keras OCR
- Google Cloud Vision API
- Microsoft Azure Cognitive Services - Computer Vision API

If the images you want to process contain tables with tabular data or other text would these tools work well for it?

The above listed packages would depend on several factors for the tabular data such as:

- Text Structure and Quality
- Preprocessing
- Support for Table Detection : Some OCR tools have specific features or models designed for table detection and extraction. For example, Google Cloud Vision API offers table recognition capabilities that can help in extracting data from tables more accurately.
- Customization and Training : Customization and fine-tuning of OCR models can significantly improve their performance for specific tasks like table extraction. Training the models with labeled data containing tables can help them better understand and extract tabular information.
- Post-Processing

If these tools/models are only 60% effective what are some things you would do to increase the effectiveness of the extraction?

Let us understand the various pre-processing tasks that would require to get the data extracted more accurately as follows:

1. Deskewing

Deskewing refers to the process of correcting skew or rotation in an image so that the text or objects in the image appear straight and aligned horizontally or vertically. This correction is essential for improving the accuracy of optical character recognition (OCR) and for visual presentation purposes.

The steps involved in deskewing are as follows and the requires packages that help us to do:

i. Detecting Skew:
   - Convert image to grayscale (cv2.cvtColor())
   - Apply edge detection or thresholding (cv2.Canny())
   - Detect lines using Hough Transform (cv2.HoughLines())

ii. Calculating Deskew Angle:
   - Calculate the dominant skew angle from detected lines. To calculate the median and degrees would use the numpy libraries such as np.median() & np.degrees()
   - The deskew angle can be calculated using algorithms that analyze the image content or by user input if the skew angle is known in advance.

iii. Deskewing Transformation:
   - After determining the deskew angle, the image undergoes a transformation to correct the skew. This transformation can be achieved using techniques such as:
     - Rotation: Rotating the image by the negative of the detected skew angle to align the text or objects horizontally or vertically.
     - Perspective Transformation: Applying a perspective transformation to correct perspective distortions caused by skew.
     - Affine Transformation: Applying an affine transformation to map points in the image to new positions, effectively straightening the skewed elements.

2. Noise Reduction:

Noise reduction refers to the process of minimizing unwanted artifacts or irregularities in an image, such as random variations in brightness or color. It aims to improve image quality by smoothing out these fluctuations without significantly affecting important details. Techniques like Gaussian blur or median filtering are commonly used for noise reduction in images, especially after deskewing or preprocessing mainly in OCR, where clean and clear images are crucial for accurate analysis.

Various steps involved in noise reduction are as follows:

i. Convert the deskewed image to grayscale.
ii. Apply a Gaussian blur filter for noise reduction.
iii. Optionally, apply thresholding to further enhance image clarity.
iv. Perform additional post-processing like contrast enhancement or adaptive thresholding if needed.

3. Contrast Enhancement
   Contrast enhancement is a technique used to improve the visual distinction between different parts of an image by increasing the difference in intensity between light and dark areas. This process makes the image appear more vivid and detailed, enhancing its overall quality and making it easier to interpret and analyze. Techniques like histogram equalization or adaptive contrast stretching are commonly used for contrast enhancement in image processing.

Various steps involved in contrast enhancement are as follows:

i. Apply contrast enhancement techniques such as histogram equalization or adaptive contrast stretching.
ii. Optionally, adjust the parameters of the contrast enhancement technique to achieve the desired level of enhancement.

Consider how you would pass along the structure/relationship between different elements of a document. How might you represent that? For instance if you pass an image into this tool what might the output contain?

When submitting an image to a tool for processing or analysis, the result may include many essential components to illustrate the organization and connections among various document elements:

- Text Extraction: Text that was extracted and arranged according to its placement and design inside the document.
- Regions of Interest (ROIs) or Bounding Boxes: Defined components such as headers, tables, paragraphs, and pictures.
- Document structure shown as a hierarchy that demonstrates the connections between sections and subsections is known as a hierarchy or tree structure.
- Extra details about an element, such as font styles, sizes, colors, and alignment, are called annotations or metadata.
- Visualizations: Diagrams, charts, or graphs that show the dependencies and relationships between the elements in the document.
