Summary of Helper Files

1. helper\_resize.py

General Description:

This module handles the preprocessing of images by identifying and extracting a yellow sticker from the original image, then resizing it for further processing.

Function Summary

- `detect\_yellow\_sticker(image\_path)`: Identifies the yellow sticker in an image by using HSV color filtering and contour detection.

- `rotate\_box(image, rect)`: Corrects the orientation of the detected sticker by rotating the image and cropping it to the sticker boundaries.

- `process\_image(image\_path)`: Main function that ties the workflow together - detects the sticker, rotates/crops it, and resizes it to a standard width of 900 pixels while preserving aspect ratio.

- `resize\_image(image, target\_width)`: Helper function that resizes any image to a specified width while maintaining its aspect ratio.

2. helper\_find\_barcodes.py

General Description

This module is responsible for detecting barcodes in different strips (regions) of the preprocessed image using adaptive thresholding and windowing techniques.

Function Summary

- `detect\_barcodes\_in\_strip(image, strip\_config, display\_results)`: Detects barcodes in a specific strip of the image using a sliding window approach with adaptive thresholding.

- `detect\_all\_barcodes(image, display\_results)`: Processes the entire image by dividing it into three strips (top, middle, bottom) and detecting barcodes in each strip.

- `draw\_bounding\_boxes\_on\_image(image, detected\_barcodes)`: Debugging function that visualizes detected barcodes by drawing colored bounding boxes around them.

Each strip is configured differently with parameters for window size, scoring thresholds, and maximum expected barcodes. The detection process combines multiple thresholding approaches and uses edge detection to identify barcode-like patterns.

3. helper\_extract.py

General Description

This module extracts relevant text fields from areas surrounding the detected barcodes using OCR (Optical Character Recognition) with Tesseract.

Function Summary

- `extract\_text\_field(image, roi\_coords, config\_str, lang)`: Extracts text from a specified region of interest using Tesseract OCR.

- `keep\_hebrew\_and\_spaces(text)`: Filters text to retain only Hebrew characters and spaces.

- `extract\_fields\_from\_barcode(image, barcode\_location, strip\_id, display)`: Extracts specific fields (name, ID, date, case ID) from regions relative to a detected barcode.

- `process\_barcodes\_and\_extract\_fields(image, barcodes\_dict, display)`: Processes detected barcodes according to a priority order and extracts fields from the highest priority barcode.

- `save\_fields\_to\_json(fields, output\_path)`: Saves the extracted fields to a JSON file.

The module uses predefined shift vectors for each strip to locate fields relative to barcode positions. It includes specialized processing for Hebrew text and numerical fields, with visualization options to display extraction regions.