Using OpenCV + PyTesseract + Python to Extract Numerical Data from Images and Export to a CSV file

In Windows, Open Irfanview and do a batch crop on all the image files you want OCR. This will reduced what is read by opencv and tesseract. In my case all my image files were the same size and data that I wanted to extract was located in a particular area. This area I would select in Irfanview, Press B (Batch Conversion), Select Batch Conversion – Rename result files. Then place a check mark next to Use advanced options. Next to it click on Advanced. Check mark Crop and Press Get current sel. Press Ok.

Now select all the image files and click on add.

Change name pattern to: ####

Then ensure Output directory for result files is in a different folder.

Open WSL

Installed Libraries
pip install opencv-python-headless numpy Pillow pytesseract
sudo apt-get install tesseract-ocr

sudo nano opencv-ocr-to-csv.py and copy and paste the code below.

Don't forget to update input_directory. Also output_file can be changed.

```
import os
import csv
import cv2
import numpy as np
import pytesseract
import re

def preprocess_image(image):

    # Convert to grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Apply thresholding to preprocess the image
    gray = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1]

    # Apply dilation to remove noise
    kernel = np.ones((1, 1), np.uint8)
    gray = cv2.dilate(gray, kernel, iterations=1)
```

```
    # Apply erosion to remove noise
    gray = cv2.erode(gray, kernel, iterations=1)

    return gray


def extract_text_from_image(image_path):
    try:

        # Read image using OpenCV
        image = cv2.imread(image_path)

        # Preprocess the image
        gray = preprocess_image(image)

        # Perform text extraction
        text = pytesseract.image_to_string(gray)
        return text.strip()

    except Exception as e:
        return f"Error processing {image_path}: {str(e)}"

def extract_numbers(text):

    # Extract all numbers (including decimals) from the text
    numbers = re.findall(r'\d+\.?\d*', text)
    return ' '.join(numbers)

def process_images(directory, output_file):
    with open(output_file, 'w', newline='', encoding='utf-8') as csvfile:
        csvwriter = csv.writer(csvfile)
        csvwriter.writerow(['Filename', 'Extracted Numbers'])  # Write header


        for filename in os.listdir(directory):
            if filename.lower().endswith(('.jpg', '.jpeg', '.png', '.bmp')):
                image_path = os.path.join(directory, filename)
                text = extract_text_from_image(image_path)
                numbers = extract_numbers(text)
                csvwriter.writerow([filename, numbers])

# Usage
```

```
input_directory = "/path/to/your/image/directory"
output_file = "extracted_numbers.csv"
process_images(input_directory, output_file)

EOF
```

Now CTRL – X to save the file.

Running the python3 script, type in the command line: python3 opencv-ocr-to-csv.py