
```

%image processing for PV measurment
%Donggeon Kim - u1419226
% Optical character recognition (OCR) from Vision toolbox of MATLAB is used
% Portions of this script were developed with the assistance of ChatGPT
  (OpenAI) and MATLAB documentation

% I= imread ('452.png');
% result = ocr(I, 'TextLayout', 'Line','characterSet','0123456789');
% outputNum=str2double(result.Text);

% My MP4 file from S25 uses the HEVC (H.265) codec, which is not supported by
  MATLAB's VideoReader.
% MATLAB supports H.264 (AVC) but not HEVC.
% Typing at this in WSL terminal -> ffmpeg -i example_30.mp4 -vcodec libx264 -
  acodec aac example_30_converted.mp4

% Define paths
videoFile = 'C:\Users\jsks12312\Desktop\PV_measurement\image_processing
\example_30_converted.mp4';

% Extract video name without extension to create a folder
[~, videoName, ~] = fileparts(videoFile);
outputFolder = fullfile('C:\Users\jsks12312\Desktop\PV_measurement
\image_processing\', videoName);

% Create output directory if it doesn't exist
if ~exist(outputFolder, 'dir')
    mkdir(outputFolder);
end

% Get current date in YYYY_MM_DD format (initialized before filenames are
  used)
currentDate = datestr(now, 'yyyy_mm_dd');

% Initialize extracted data structure before the loop
extractedData = struct('Time', {}, 'ExtractedText', {}, 'ImageFile', {});

% Open the video file
video = VideoReader(videoFile);

% Get the total duration of the video
videoDuration = min(10, video.Duration); % Process only up to 10 sec

% Create a file to save extracted OCR results in the same folder
ocrResultsFile = fullfile(outputFolder, sprintf('%s_
%s_Extracted_OCR_Results.txt', currentDate, videoName));
fileID = fopen(ocrResultsFile, 'w'); % Overwrite previous results
fprintf(fileID, 'Time (s)\tExtracted Text\n');

% Define capture interval within each second ( $\pm 0.1$ s around the target time)
captureOffset = 0.1;

```

```

% Iterate over the video from 0 to 10 seconds
for targetTime = 0:1:videoDuration % Iterate at 1s intervals
    % Generate timestamps, ensuring no negative values
    timeStamps = max(0, [targetTime - captureOffset, targetTime, targetTime +
captureOffset]);

    % Iterate over the three timestamps per second (0.0s, 0.1s, 0.2s, etc.)
    for i = 1:length(timeStamps)
        % Set the video to the desired timestamp
        video.CurrentTime = timeStamps(i);

        if hasFrame(video)
            frame = readFrame(video);

            % Convert seconds to HH_MM_SS_MMM format (precise video time with
milliseconds)
            totalMilliseconds = round(video.CurrentTime * 1000); % Convert
seconds to milliseconds
            hours = floor(totalMilliseconds / (3600 * 1000));
            minutes = floor(mod(totalMilliseconds, (3600 * 1000)) / (60 *
1000));
            seconds = floor(mod(totalMilliseconds, (60 * 1000)) / 1000);
            milliseconds = mod(totalMilliseconds, 1000);

            % Format timestamp as "05h_20m_02s_100ms"
            % to meet the format
            timeStr = sprintf('%02dh_%02dm_%02ds_%03dms', hours, minutes,
seconds, milliseconds);

            % save full frame inside the video-named folder for
            % organization
            frameFilename = fullfile(outputFolder, sprintf('%s_%s_frame_
%s.png', currentDate, videoName, timeStr));
            imwrite(frame, frameFilename);
            disp(['Saved full frame at ',
num2str(video.CurrentTime, '%.3f'), ' sec as ', frameFilename]);

            % Convert to grayscale and preprocess to help image
            % recognition
            grayFrame = rgb2gray(frame);
            grayFrame = imadjust(grayFrame, [0.3, 0.7], [0, 1]); % enhance
contrast
            grayFrame = medfilt2(grayFrame, [3, 3]); % Reduce noise

            % select region of interest from images
            if targetTime == 0 && i == 1 % Only ask for ROI once
                figure, imshow(frame);
                title('Select ROI for Multimeter Display');
                roi = round(getrect); % User selects ROI
                disp(['Selected ROI: ', num2str(roi)]);
            end

            % Crop the frame to the ROI - it works at being saved at

```

```

        % desired location
        croppedFrame = imcrop(grayFrame, roi);

        % Save cropped frame in the same folder with precise timestamp
        croppedFilename = fullfile(outputFolder, sprintf('%s_%s_cropped_
%s.png', currentDate, videoName, timeStr));
        imwrite(croppedFrame, croppedFilename);
        disp(['Saved cropped image at ',
num2str(video.CurrentTime, '%.3f'), ' sec as ', croppedFilename]);

%failing point
    % apply OCR to the cropped frame
    result = ocr(croppedFrame, 'CharacterSet', '0123456789.-');
    extractedText = strtrim(result.Text);

    % display extracted text on terminal
    disp(['Extracted Text at ', num2str(video.CurrentTime, '%.3f'), '
sec: ', extractedText]);

    % save extracted text to file
    fprintf(fileID, '%.3f\t%s\n', video.CurrentTime, extractedText);

    % Store extracted data
    extractedData(end+1) = struct( ...
        'Time', video.CurrentTime, ...
        'ExtractedText', extractedText, ...
        'ImageFile', croppedFilename);
    end
end
end

% Close results file
fclose(fileID);

% Convert extracted data to a table and display in MATLAB
% it needs work
if ~isempty(extractedData)
    extractedTable = struct2table(extractedData);
    disp(extractedTable);
    % Save extracted data to CSV in the same folder
    csvFilename = fullfile(outputFolder, sprintf('%s_
%s_Extracted_OCR_Data.csv', currentDate, videoName));
    writetable(extractedTable, csvFilename);
    disp(['OCR results saved to ', csvFilename]);
else
    disp('No text was extracted from the video.');
```

end

```

disp('OCR Processing Complete.');
```

*Saved full frame at 0.033 sec as C:\Users\jsks12312\Desktop
\PV_measurement\image_processing\example_30_converted
\2025_03_22_example_30_converted_frame_00h_00m_00s_033ms.png*

```
Error using getrect (line 167)
Interruption during mouse selection.

Error in image_processing (line 87)
    roi = round(getrect); % User selects ROI
```

Published with MATLAB® R2021b