METHODOLOGY

The proposed scheme consists of several sub modules as shown in the flow diagram in Fig.1. All the modules of the proposed scheme have been written in Python 3.9.

The code is provided here: https://github.com/CaffineAddic/Wavelet_transforms_in_schlieren_flame_images

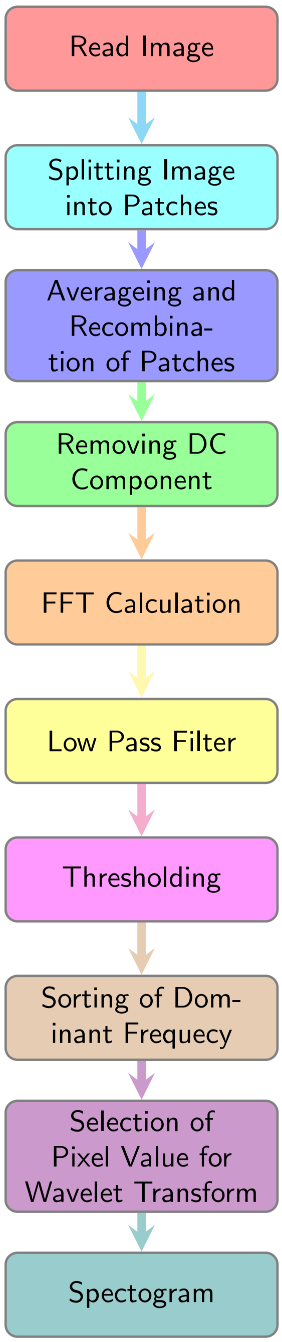

Fig. 1. Flowchart of the image processing procedure

An exemplary instantaneous Schlieren image of the liquid spray from the high-speed image sequence is shown in Fig. 2

Algorithm 1 FFT Analysis and Visualization

Input: $\{P_i\}_{i=1}^{N}$, a set of N input data points
Output: $\{F_i\}_{i=1}^{M}$, a set of M output frequency distributions
procedure DATAPREPROCESSING
  Load input data: $\{P_i\}_{i=1}^{N}$
  Select time range: $T = [t_{start}, t_{end}]$
  Crop data: $D = \{P_i\}_{i=1}^{N} \cap T$
end procedure
procedure PATCHIFICATION
  Create patches: $Patches = \{P_i\}_{i=1}^{N} \mapsto \{Patches_i\}_{i=1}^{N}$
  Calculate mean of patches: $\{Mean_i\}_{i=1}^{N}$
end procedure
procedure FFT
  Calculate FFT of mean: $\{FFT_i\}_{i=1}^{N}$
  Filter FFT: $\{Filtered_i\}_{i=1}^{N}$
end procedure
procedure FREQUENCYDISTRIBUTION
  Calculate frequency distribution: $\{F_i\}_{i=1}^{M}$
  Sort and arrange data: $\{F_i\}_{i=1}^{M} \mapsto \{Sorted_i\}_{i=1}^{M}$
end procedure
procedure VISUALIZATION
  Plot frequency distribution: $\{Sorted_i\}_{i=1}^{M} \mapsto \{Plot_i\}_{i=1}^{M}$
  Save plots: $\{Plot_i\}_{i=1}^{M} \mapsto \{File_i\}_{i=1}^{M}$
end procedure

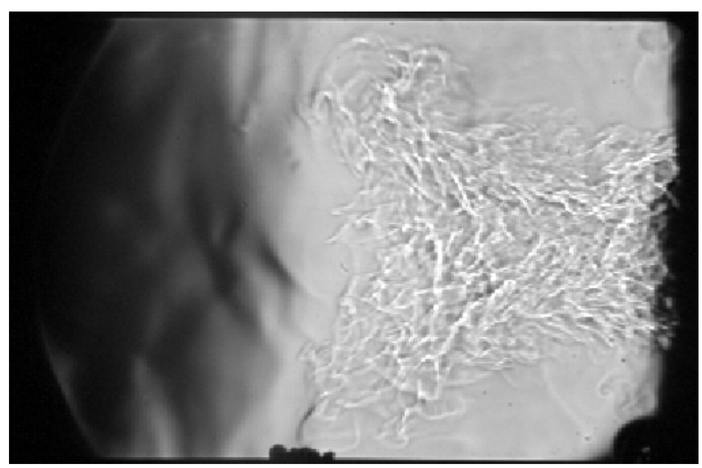

Fig. 2. Image obtained from Schlieren imaging

A. *Patches Splitting and Recombination after Averaging*

The Schlieren images are high resolution to reduce computational overhead and time, the images are split into small image patches, further in each patch the average value of the entire patch is calculated and integrated together to form pooled images with a lower resolution while preserving information.

A sample image after the process is depicted in Fig. 3. Here, the image has been color-mapped to a specific scheme and is presented with a color bar, allowing for a visual representation of pixel intensity.

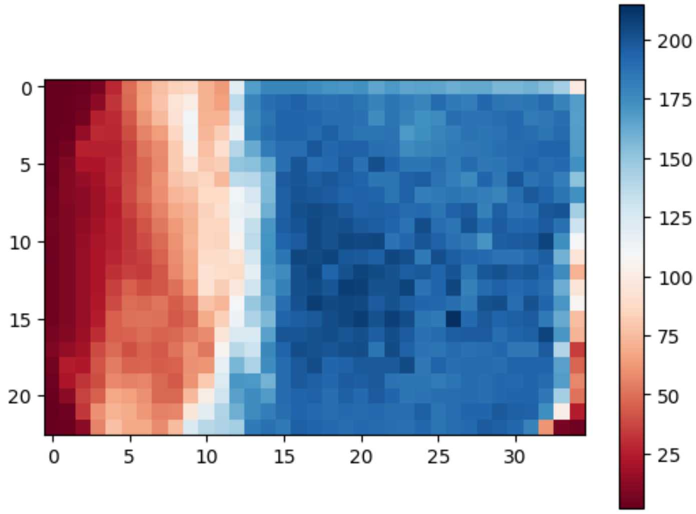

Fig. 3. Image obtained after integrating the average of patches

B. DC component removal & Fast Fourier Transform (FFT) calculation

For each post-integrated image in the sequence, the average of all pixel values is computed and subtracted from each pixel to eliminate the DC component Fig. 4. This process, known as centering the data or mean normalization, helps remove bias in the dataset. After applying mean normalization to all images in the series, the FFT is calculated for each pixel, spanning across the entire integrated image sequence.

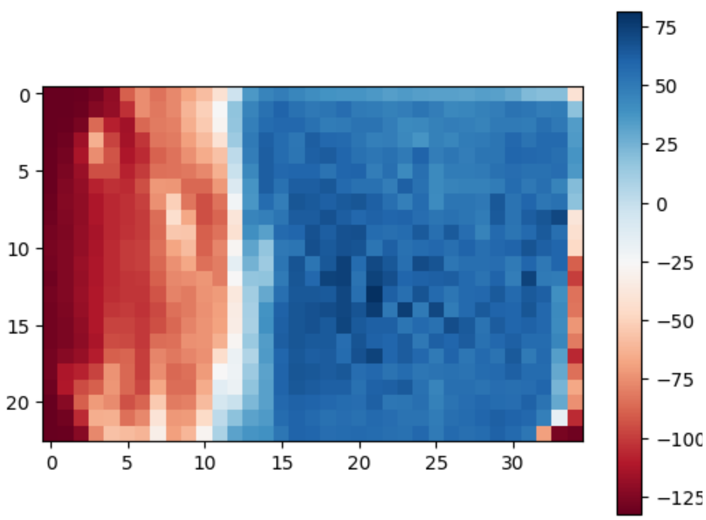

Fig. 4. Image after mean normalization

C. Low Pass Filtering (LPF)& Thresholding

LPT filtering is utilized on the FFT series to eliminate noise, particularly targeting the low-frequency components. Additionally, a thresholding process based on the FFT amplitude is implemented to filter out frequency components with low amplitudes. This dual approach helps to enhance the quality of the FFT data by reducing noise and focusing on significant frequency components.

D. Sorting of Dominant Frequencies

The frequency values in the FFT array are sorted according to their FFT amplitudes. Subsequently, the highest amplitude value from each sorted FFT array is integrated into an image to visualize the most dominant frequency for each pixel. This process allows for the observation of the primary frequency components across the image, as illustrated in Fig. 5. Further the FFT amplitude of each dominating frequency is shown in Fig. 6.

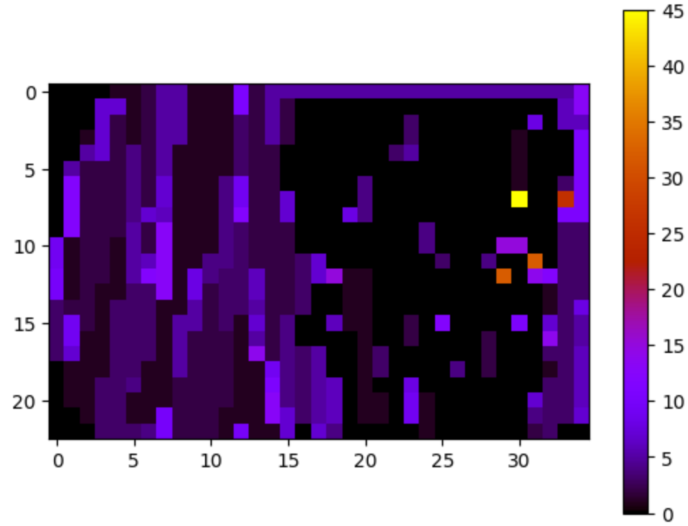

Fig. 5. Image displaying all the dominant frequency for each pixel

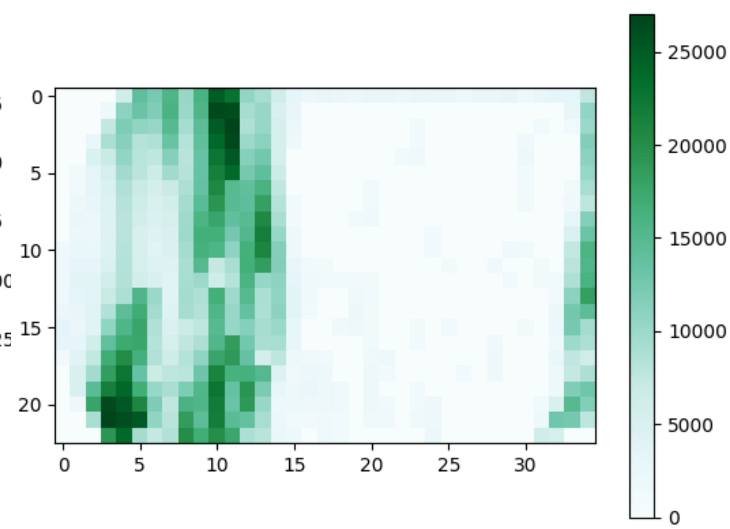

Fig. 6. Corresponding amplitude for each frequency in Fig. 5

E. Wavelet Transform Calculation and Spectrogram

From the dominant frequency flame image, a specific pixel of interest is chosen to extract temporal information of the frequencies present. This involves identifying the time duration and the frequencies displayed by the selected pixel. This targeted approach allows for a detailed analysis of the temporal characteristics and frequency components exhibited by the pixel under investigation. Further spectrogram is plotted to display that information as in Fig. 7.

Algorithm 2 Wavelet Analysis of Pixel-Selected Intensity Time Series

Input: pl (Pixel location) [User Input]
$s = data[:, pl[0], pl[1]]$
$t = gen\_time(data.shape[0])$
$t = t/fs + init\_time$
$sc = gen\_scale(min\_scale, max\_scale)$
$c, f = wt(s, sc, wavelet, fs)$
Plot scalogram using t, f, and c with 'inferno' colormap
Save to file 'calo-' + pl + '.png'

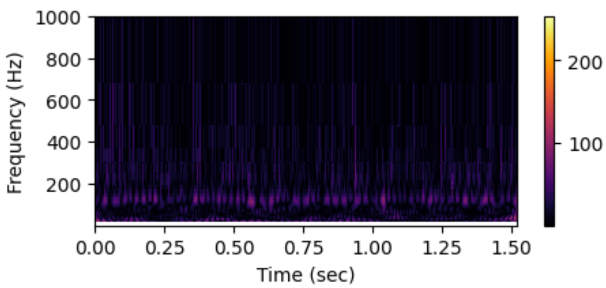

Fig. 7. The spectrogram of the pixel of interest.