Code process flow

The code utilizes patchify (to split the images in patches), numpy (to add array toolbox), scipy (for FFT calculations), pywt (to calculate the wavelet transfrom) and matplotlib (to plotting the values).

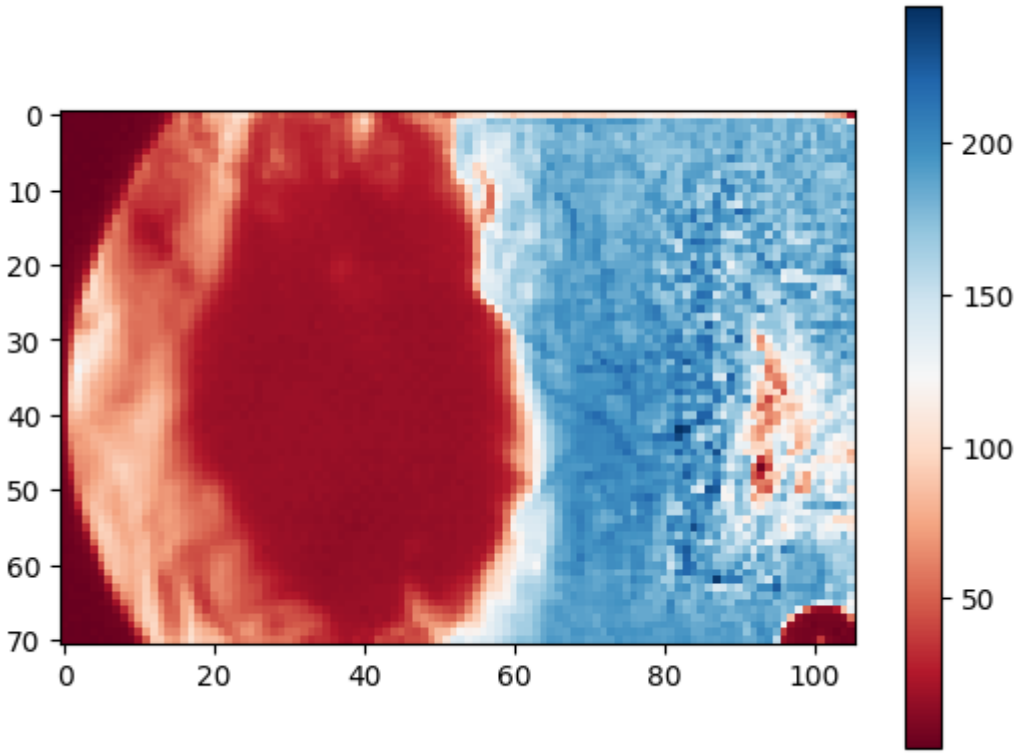
Images taken with a high speed camera and the each image corresponds to a temporal value.

The code starts by reading the images specified by the user and use patchfy to reduce the image size by pooling spliting images into small overlappable patches by given patch cell size, and merge patches into original image using.

```
bin = patch cell size
```

After the image is split into patches the average value of each patch is taken and used to form a complete pooled image with reduced dimensions given by the binning.

Once a complete pooled image is formed the average of all the pixels in the binned image is taken and subtracted from each pixel to remove the DC component.



Binned image sample

Using scipy the FFT of the signal is calculated over each pixel value in the image across all the temporal binned images.

From the FFT values of all the pixels, the low frequency are filtered as the cutoff defined by the user.

```
cutoff = low frequency cutoff values
```

From all the fft values in each pixel all the low amplitude frequencies are eliminated as defined by the user, by thresholding the frequency values in FFT in percentage range of the maximum amplitude present in the FFT array of each pixel.

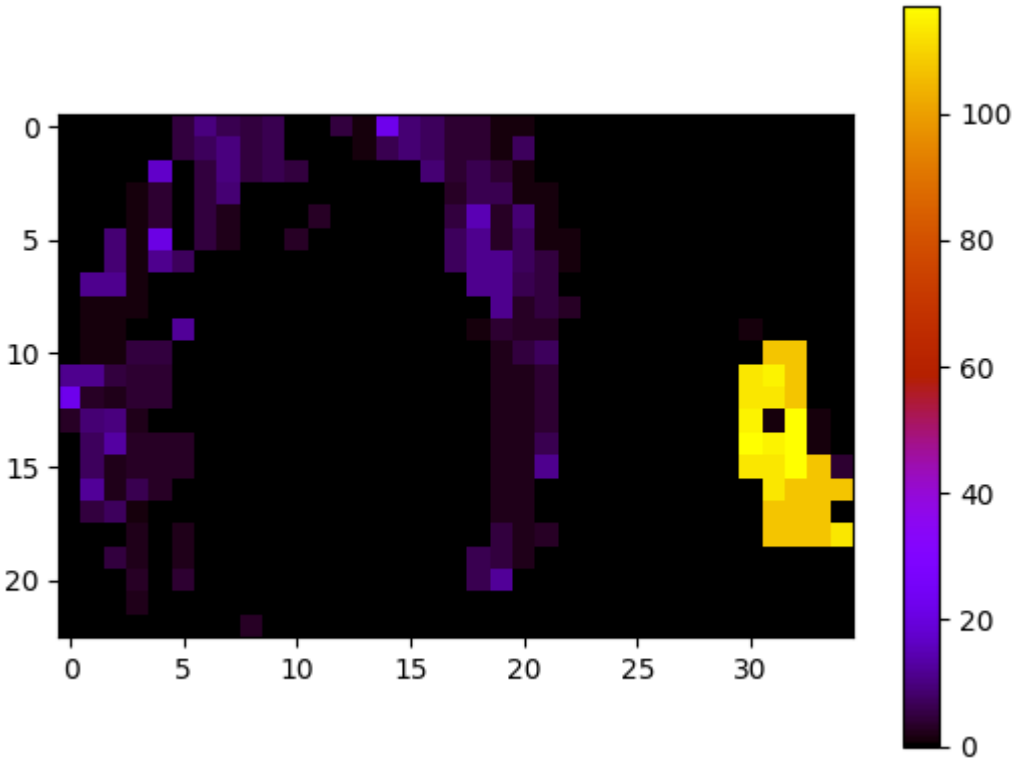
```
threshing = % range in the aplitude with respect to the max value
```

for example if it is .9, the top 90% values based on aplitiude will left and the smallest 10% of the values will be eliminated.

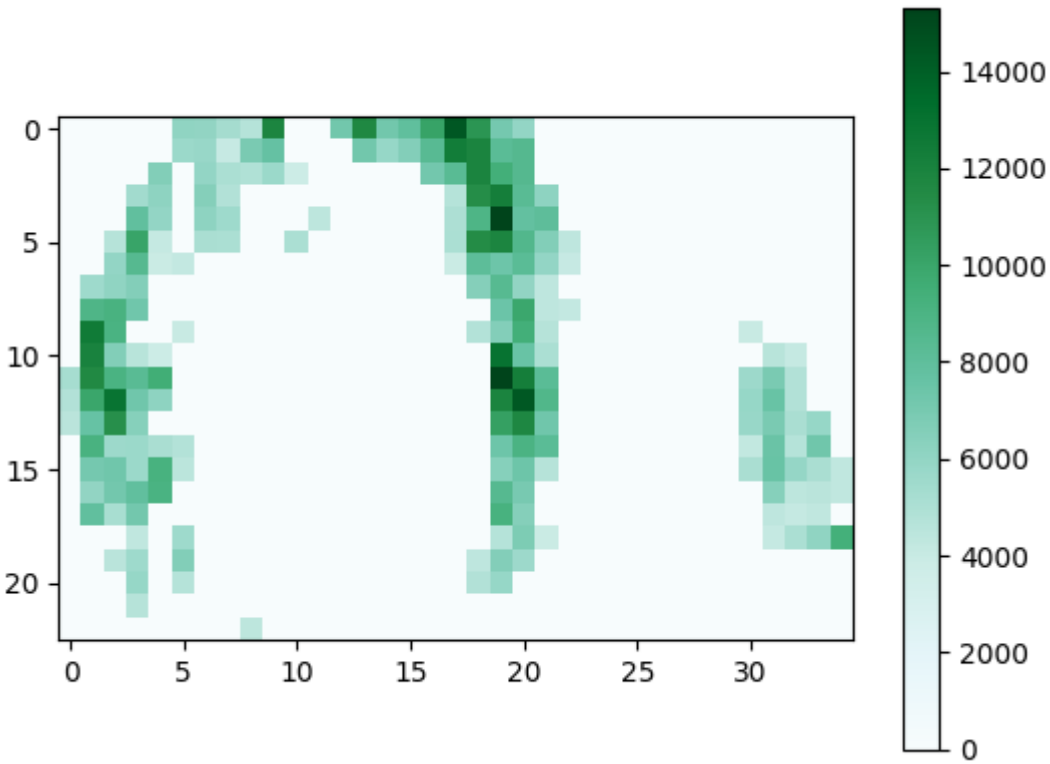
After thesholding the all the frequency value present in the FFT are sorted based on their amplitude i.e. the frequency with biggest amplitude will be the first in the array. This done for all the pixel array FFTs.

This allowed all arrays to be in order of their dominating frequency for example if we plot the first values of the array together we will have an image of flame where we are shown the most dominating frequency in particular pixel.

Two images are plotted 1st showing their dominant frequency and the 2nd one showing their corresponding amplitudes.

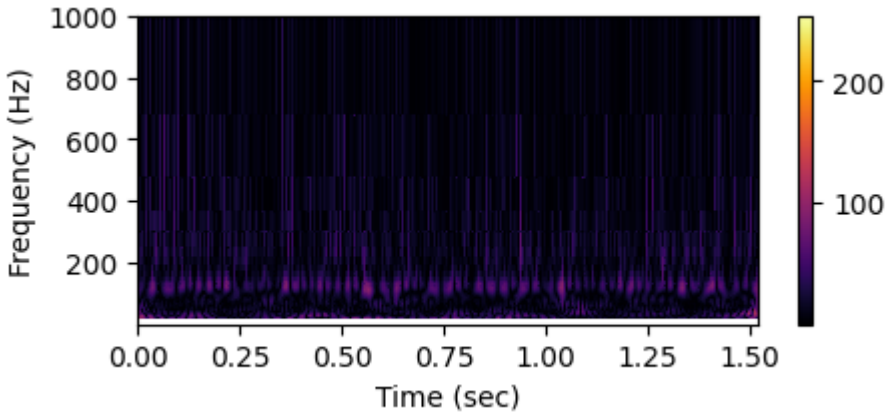


1st image



2nd image

Now we use pywt to see wavelet transform of the selected pixel to observe how the frequencies were present in the particular pixel with respect to time.



pywt output

Further more here the time interval can be selected to observe values in that time frame.