

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W
KRAKOWIE
WYDZIAŁ INFORMATYKI



Przetwarzanie sygnałów medycznych

Ćwiczenia laboratoryjne - Informatyka Medyczna

Paulina Dziwak
Piotr Ludynia

Kraków, 2024

Podczas zajęć eksplorowaliśmy innowacyjną koncepcję pozyskiwania informacji na temat interakcji końcówki igły z tkanką podczas przezskórnego wprowadzania igły. W przeciwieństwie do tradycyjnych metod obrazowania, które mogą być obciążone niedoskonałościami wynikającymi z artefaktów, podejście oparte na dźwięku ma potencjał dostarczenia większej ilości informacji.

Zdecydowaliśmy się pominąć opisy zadania związane z wczytywaniem plików w naszym sprawozdaniu ze względu na ich prostotę. Pliki te znajdują się w katalogu "data", obok skryptu odpowiedzialnego za przeprowadzanie obliczeń.

Zadanie 1. Filtracja Gaussowska klatek:

Podczas tego zadania zastosowaliśmy filtr Gaussowskiego z parametrem $\sigma = 1.0$. Taka wartość parametru pozwala na uzyskanie większego rozmycia obrazu niż w przypadku niższych wartości filtra.

```
frame_filtered = [gaussian_filter(frame, sigma=1.0) for frame in self.frames]
```

Zadanie 2. Bandpass Filtering

Implementacja butterworth bandpass pozwala na sprecyzowanie częstotliwości na 2 sposoby:

- Pierwsza metoda polega na znormalizowaniu wartości granicznych (górnej i dolnej) przy pomocy częstotliwości próbkowania. Zgodnie z teorią przetwarzania sygnałów, maksymalna częstotliwość, którą można przetworzyć filtrem, wynosi połowę częstotliwości próbkowania ($0.5 \times fs$).

```
nyq = 0.5 * fs  
b, a = butter(order, [lowcut/nyq, highcut/nyq], btype='bandpass', analog=False)
```

- Druga metoda wykorzystuje bibliotekę `scipy.signal`, która pozwala na przekazanie wartości częstotliwości próbkowania (sampling frequency) do filtra. Dzięki temu nie ma potrzeby normalizowania punktów odcięcia.

```
b, a = butter(order, [lowcut, highcut], btype='bandpass', analog=False, fs=fs)
```

Zadanie 3. Mediana

Biblioteka NumPy pozwala nam bardzo łatwo obliczyć medianę z klatek:

```
self.frame_median = np.median(self.frames_filtered[start_idx:end_idx], axis=0)
```

Zadanie 4. Różnica między klatką a medianą

W celu dodatkowego przefiltrowania obrazu odjęliśmy od klatek filmu medianę wartości sąsiadujących klatek, aby lepiej wyróżnić zachodzące zmiany na obrazie USG.

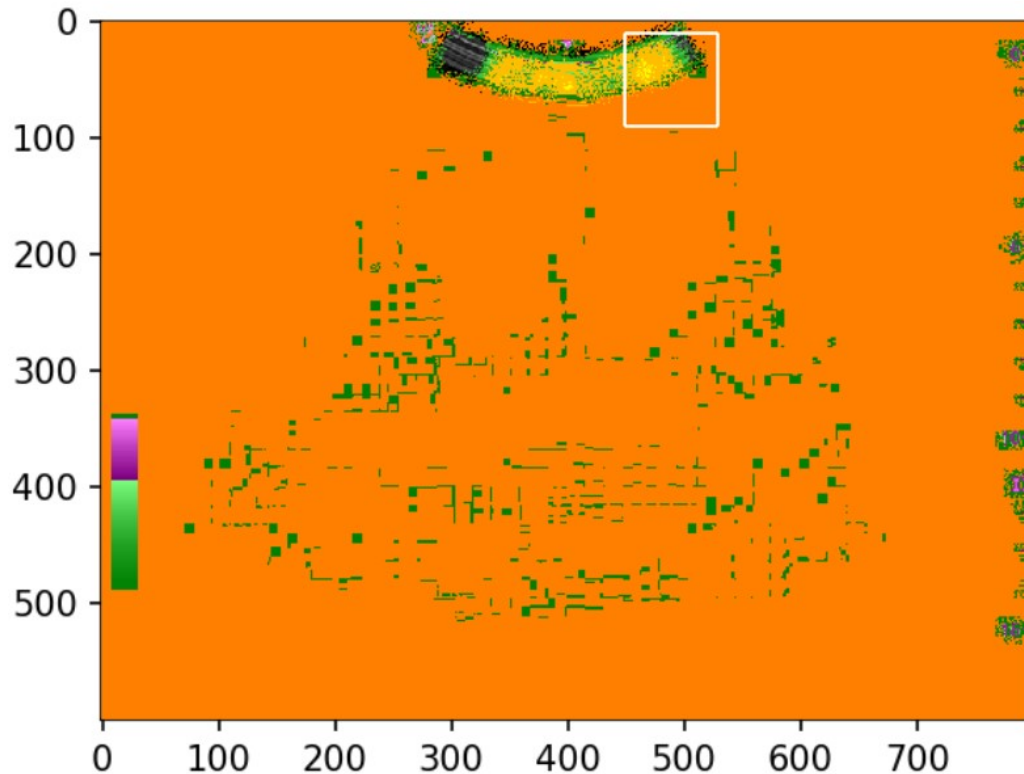
- Eksperymentalnie okazało się, że łatwiej jest zrozumieć, co dzieje się na obrazie, jeśli użyjemy wartości bezwzględnej. Pozwala to nie tylko na znalezienie wartości, które się pojawiają, ale także tych, które znikają. Dzięki temu możemy wykryć igłę zarówno w momencie jej pojawienia się, jak i zniknięcia.

```
# todo: calculate the difference between the filtered frames and the median
# hint: normalize the frame values afterwards
frame_diff_median = np.abs(frame_filtered - self.frame_median)
```

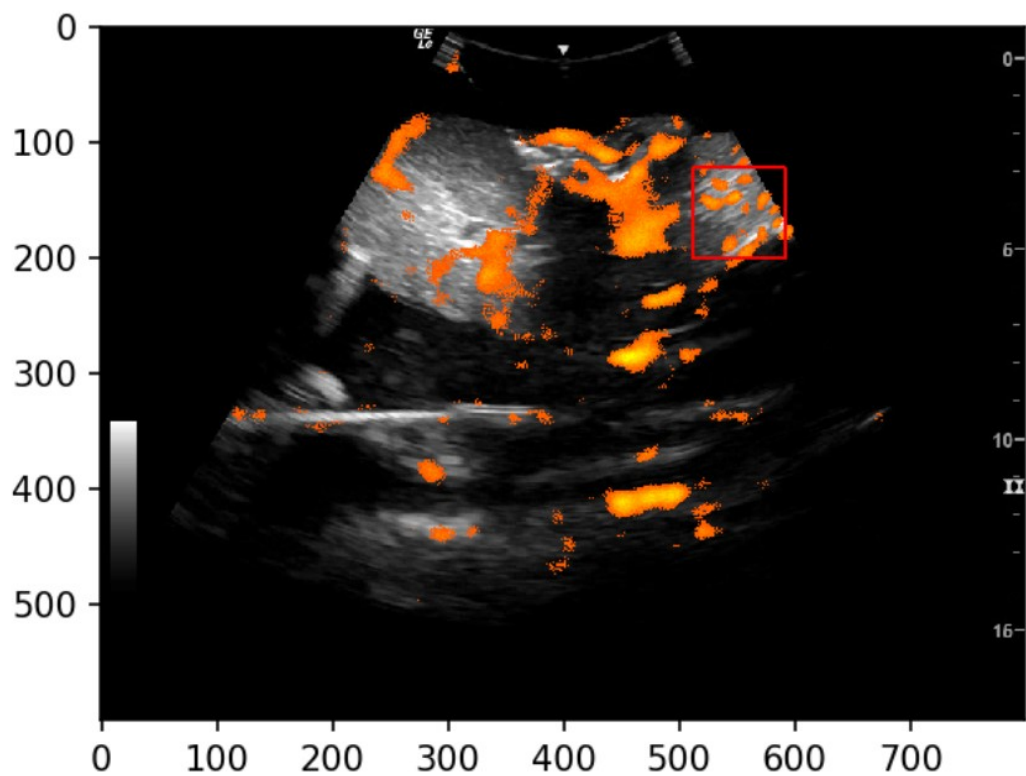
- Następnie wartości zostały znormalizowane do zakresu od 0 do 255 i skonwertowane do typu unsigned int8.

```
frame_diff_median = (255 * frame_diff_median / frame_diff_median.max())
.astype(np.uint8)
```

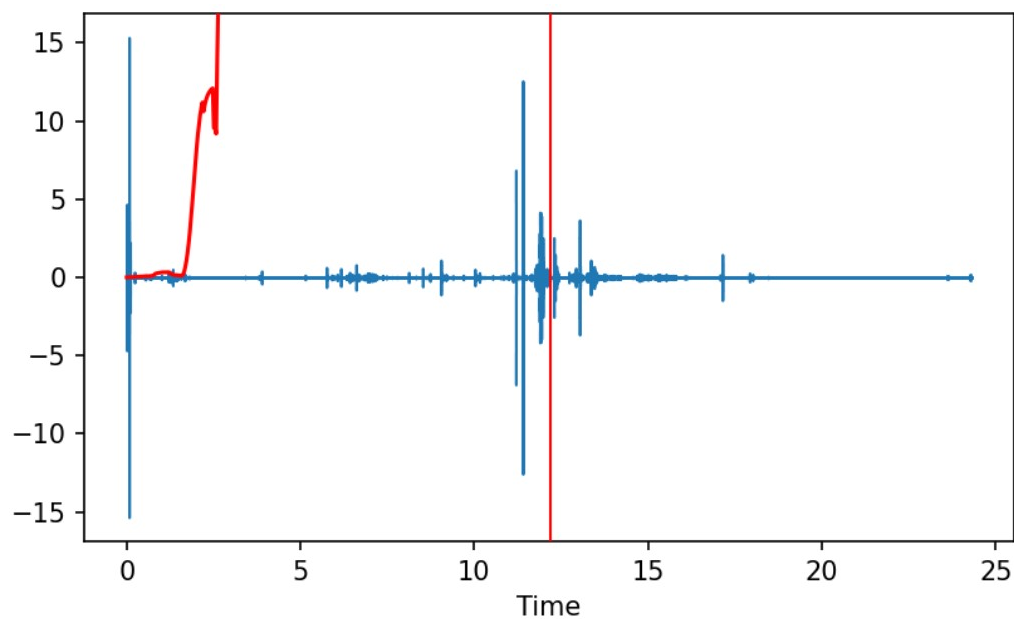
- Próba normalizacji wartości bez użycia wartości bezwzględnej prowadziła do uzyskania niechcianych rezultatów:



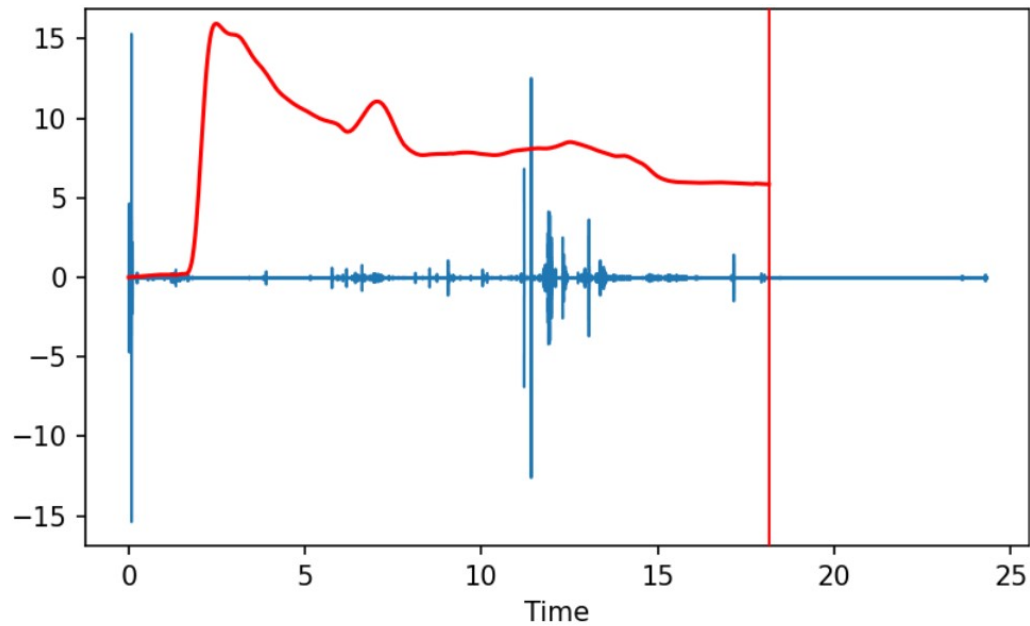
W trakcie tego etapu eksperymentu, ze względu na znaczące zmiany wartości na całym obrazie, okno deskryptora przesuwało się w obszary, które nie były naszym głównym punktem zainteresowania.



Spowodowało to uzyskanie wyników wykraczających daleko poza oczekiwany zakres wartości.



Po ustaleniu pozycji okna deskryptora na stałej pozycji (310,50), uzyskaliśmy wykres obrazujący pewną korelację między danymi.



Pierwszy wzrost wartości deskryptora jest spowodowany włączeniem kamery i uruchomieniem działania programu. Na filmie pojawia się wtedy dużo nowych wartości. W momencie, kiedy na obrazie igła jest najbardziej widoczna (drugie lokalne maksimum na wykresie), widać sygnał akustyczny.



W momencie przebicia igły przez tkanę, na filmie nie pojawia się bardzo dużo nowej informacji, więc wartość deskryptora wzrasta tylko nieznacznie.

Wnioski:

Zadanie polegające na znalezieniu odpowiedniego sposobu umiejscowienia okna deskryptora wydaje się być niezwykle trudne. Można by zaproponować rozwiązanie machine learningowe i przeprowadzić regresję by przewidzieć wartość sygnału dźwiękowego na podstawie obrazu.