

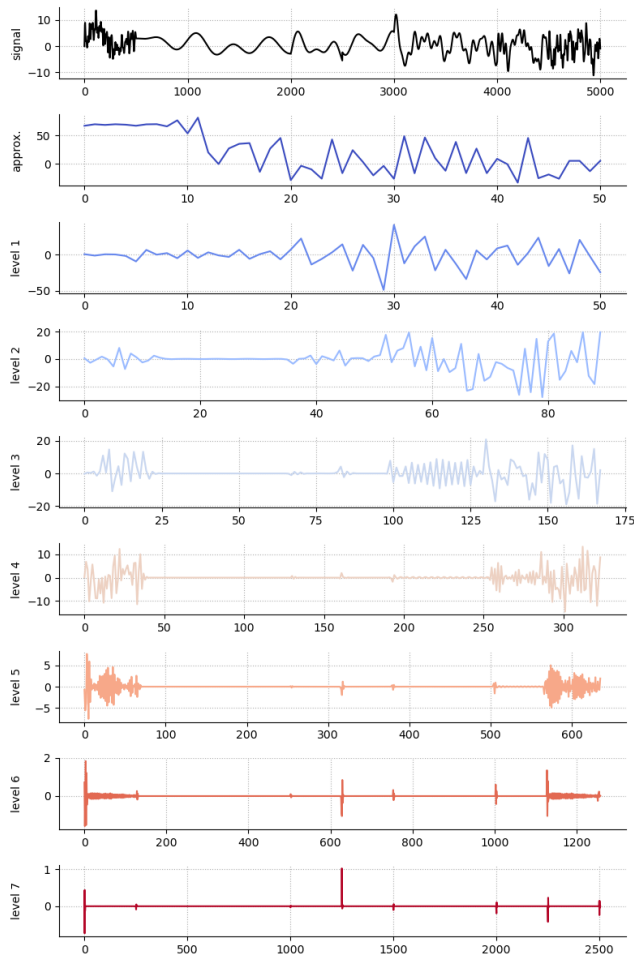
Laboratorium 7

Słowa kluczowe: transformata falkowa, wielorozdzielczość, filtracja.

Zadanie 1:

- Wczytaj sygnał znajdujący się w pliku `signal.npy`.
- Wykonaj wielopoziomową dyskretną transformatę falkową tego sygnału, wykorzystując funkcję `wavedec` z biblioteki `PyWavelets` (`pywt`).
 - wykorzystaj do tego falkę `db7` z rodziny *Daubechies*.
 - sprecyzuj poziom (`level`) na wysokość 7.
- Na wykresie przedstaw:
 - sygnał oryginalny,
 - współczynniki przybliżone (*approximation coefficient*) — pierwszy obiekt zwracany przez funkcję `wavedec`,
 - współczynniki szczegółowe (*detail coefficient*) z 7 kolejnych poziomów.
- Zadbaj o stylowanie wykresu:
 - ustaw siatkę (`grid`),
 - podpisz sygnały na osi y (`set_ylabel`),
 - wykorzystaj spójne kolory np. próbkowanie colormapy:
`plt.cm.coolwarm(parametry z zakresu 0-1)`,
 - ukryj oś po prawej stronie i nad wykresem (`spines['top'].set_visible(False)`),
 - wyrównaj podpisy osi (`align_ylabels`),
 - zastosuj `tight_layout`.

Efekt zadania 1:



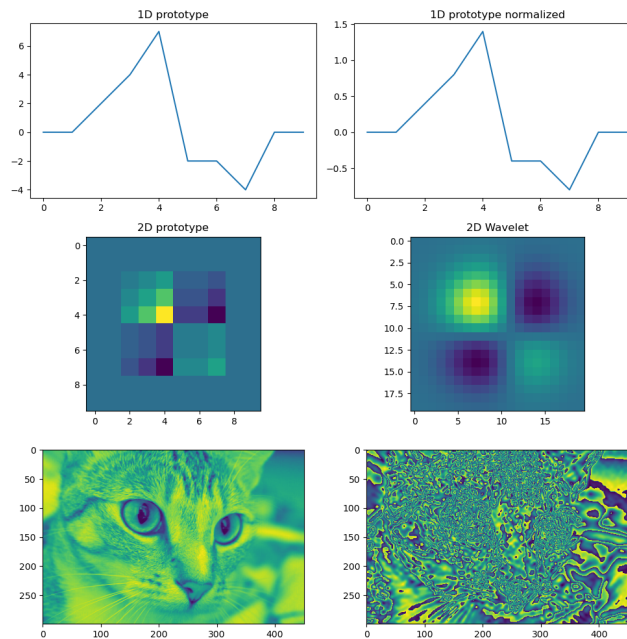
Zadanie 2:

W tym zadaniu zajmiemy się już sygnałem *wielowymiarowym* — znanym już obrazem chelsea. Każdy z Państwa będzie filtrować ten sygnał *osobistą falką* utworzoną na podstawie numeru indeksu.

- Opracuj osobistą falkę dwuwymiarową:
 - Utwórz prototyp falki na podstawie numeru indeksu. Dla numeru indeksu $abcdef$ utwórz wektor: $w = [0, 0, a, b, c, -d, -e, -f, 0, 0]$.
 - Znormalizuj wektor tak, aby **suma** wartości wynosiła 1.
 - Opracuj złożenie tego prototypu do dwóch wymiarów: `w[:,None]*w[None,:]`. Wynik powinien być dwuwymiarowy o kształcie (10,10).
 - Zmień rozmiar otrzymanego prototypu na (20,20) wykorzystując funkcję `resize` z biblioteki `scikit-image`.
 - Wykonaj filtrację za pomocą filtra gaussowskiego z parametrem `sigma` równym 2. Wykorzystaj w tym celu gotową funkcję `gaussian_filter` z biblioteki `scipy`.
- Na wykresie (3 wiersze, 2 kolumny) przestaw kolejno:
 - oryginalny prototyp jednowymiarowy na podstawie numeru indeksu,
 - znormalizowany prototyp jednowymiarowy,
 - złożenie dwuwymiarowe prototypu,

- otrzymaną dwuwymiarową falkę — prototyp dwuwymiarowy po powiększeniu i filtracji.
- W końcu przejdźmy do filtracji obrazu.
 - Wczytaj obraz *chelsea*, uśrednij kanały barwne.
 - Oblicz korelację obrazu *chelsea* z opracowaną falką, wykorzystując w tym celu gotową funkcję `correlate` z biblioteki `scipy`.
 - W ostatnim wierszu wykresu przestaw obraz po uśrednieniu kanałów oraz po filtracji falkowej.

Efekt zadania 2:



Zadanie 3:

Typowe dla transformaty falkowej jest *skalowanie* falki, aby odszukać różnorodne cechy częstotliwościowe obrazu. W tym zadaniu będziemy wyszukiwać składowe na 16 poziomach.

- Wykorzystaj w tym zadaniu obraz *chelsea* z poprzedniego zadania (po uśrednieniu kanałów barwnych, ale przed filtracją) oraz *osobistą falkę* 2D (tą o rozmiarze 20,20).
- Przygotuj wykres (4 wiersze, 4 kolumny). Tutaj przyda się funkcja `np.ravel` do zmiany kształtu listy `ax`.
- Zadeklaruj rozmiary falek — będzie to 16 wartości typu całkowitoliczbowego, równo próbkowanych od 2 do 32.
- Dla każdego zadanego rozmiaru *s* falki:
 - zmień rozmiar falki do rozmiaru *s* (funkcja `resize`),
 - wykonaj filtrację (`correlate`) obrazu oryginalnego za pomocą przekształconej falki.
 - wynik zaprezentuj w odpowiedniej komórce wykresu. W opisie komórki umieść aktualny rozmiar *s*.
- Opcjonalnie: odzyskaj *najciekawszą* colormapę dla otrzymanych efektów filtracji.

Efekt zadania 3:

