

Laboratorium 6

Słowa kluczowe: obrazy barwne, obrazy nadwidmowe, ekstrakcja cech, klasyfikacja.

Zadanie 1:

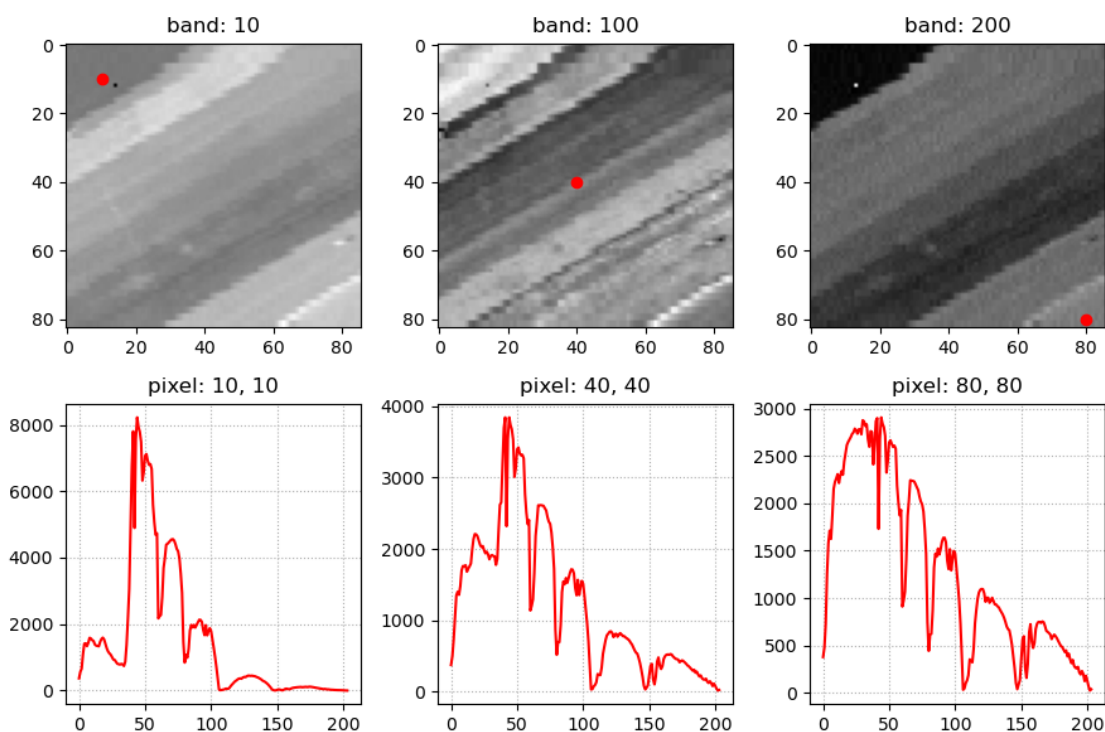
- Wczytaj obraz nadwidmowy `SalinasA_corrected.mat` w następujący sposób:

```
scipy.io.loadmat('SalinasA_corrected.mat')['salinasA_corrected']
```

Wczytany poprawnie obraz powinien mieć kształt (83, 86, 204).

- Przygotuj wykres zawierający trzy kolumny i dwa wiersze.
- W pierwszym wierszu zaprezentuj intensywność w 10., 100. i 200. paśmie głębi spektralnej obrazu (oś x).
- W drugim wierszu narysuj sygnatyry spektralne pojedynczych pikseli obrazu: punktów na współrzędnych (10,10), (40,40) oraz (80,80).

Efekt zadania 1:



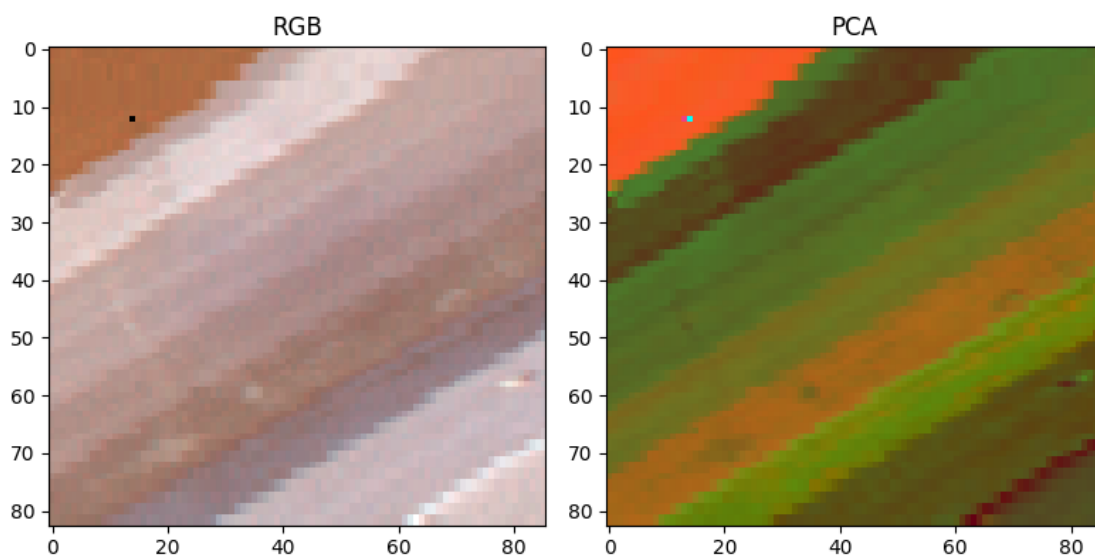
Zadanie 2:

Akwizycji danych do obrazu SalinasA dokonano za pomocą sensora AVIRIS, pobierającego 224 pasma długości fali z zakresu 400 - 2500 nanometrów (później część usunięto, stąd mamy ich 204).

Można obliczyć, że pasma odpowiadające barwom czerwonej, zielonej i niebieskiej znajdują się na indeksach: 4, 12, 26.

- Wybierz te pasma, które odpowiadają odpowiednim składowym RGB, następnie dokonaj normalizacji przedziałowej w obrębie każdego z trzech nowych kanałów barwnych.
- Wynik (obraz barwny — złożenie RGB) zaprezentuj na wykresie.
- Wykonaj ekstrakcję atrybutów metodą PCA do 3 komponentów. Wykorzystaj w tym celu gotową implementację PCA z biblioteki `scikit-learn`.
 - Tutaj zaczynamy traktować piksele obrazu jak **wzorce** w uczeniu maszyn. Co oznacza, że należy przekształcić reprezentację obrazu do kształtu: liczba pikseli x liczba pasm. Piksele staną się instancjami opisywanymi przez 204 cechy — odczyty z kolejnych pasm. Taka reprezentacja przed PCA powinna mieć kształt (7138, 204), a po PCA (7138, 3).
 - Po ekstrakcji cech należy wynik przekształcić z powrotem do kształtu obrazu barwnego (83, 86, 3).
 - Nowe kanały barwne należy również znormalizować przedziałowo.
- Wynik (obraz barwny — komponenty z PCA) zaprezentuj na wykresie.

Efekt zadania 2:



Zadanie 3:

Celem ostatniego zadania jest porównanie jakości klasyfikacji dla różnych reprezentacji obrazu:

1. po selekcji kanałów RGB,
2. po ekstrakcji cech metodą PCA,
3. dla pełnej głębi spektralnej (wszystkie cechy).

- Wczytaj plik `SalinasA_gt.mat` zawierający etykiety:

```
scipy.io.loadmat('SalinasA_gt.mat')['salinasA_gt']
```

Obraz zawierający etykiety powinien mieć początkowo kształt (83, 86), ale możemy od razu go przekształcić do wektora o długości 7138.

- Przejdź ponownie do reprezentacji obrazów typowej dla uczenia maszyn — *liczba wzorców x liczba cech*.
- Zignoruj wzorce i etykiety tła (piksele dla których etykieta wynosi 0) — tutaj przyda się *maskowanie*.
- W pętli *powtórzonej stratyfikowanej walidacji krzyżowej* (5 powtórzeń x 2 foldy) przetestuj działanie swojego ulubionego klasyfikatora z biblioteki `scikit-learn` dla trzech badanych reprezentacji danych. Wykorzystaj gotową klasę `RepeatedStratifiedKFold` do podziału zbioru na treningowy i testowy.
- Uśrednione wyniki eksperymentu oraz ich odchylenia standardowe (z dokładnością do trzech miejsc po przecinku) wypisz w terminalu.

Przykładowy efekt zadania 3 (dla GNB):

```
RGB 0.911 (0.003)
PCA 0.935 (0.003)
All 0.888 (0.005)
```