

SPRAWOZDANIE

Zajęcia: Matematyka Konkretna

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 2 Data 05.04.2025 Temat: Metoda SVD w celu obliczenia korelacji Wariant 6	Imię Nazwisko Hubert Mentel Informatyka II stopień, niestacjonarne, 2 semestr, gr.1a
--	---

1. Cel:

Celem jest nabycie podstawowej znajomości użycia rozkładu SVD w celu rozwiązywania problemów powiązanych z użyciem korelacji.

2. Zadania:

Opracować rekurencyjną sieć neuronową która implementuje operacje na dwóch liczbach binarnych zgodnie z wariantem zadania:

Wariant 6 - Suma dwóch liczb 24-bitowych.

Zadanie dotyczy obliczenia korelacji obrazu zgodnie z wariantem pod względem zarówno wierszy jak i kolumn z użyciem SVD. Przedstawić macierzy korelacji graficznie. We wniosku potraktować widoki macierzy korelacji.

Pliki dostępne są pod linkiem:

<https://github.com/HubiPX/NOD/tree/master/MK/Zadanie%202>

3. Opis programu opracowanego (kody źródłowe, zrzuty ekranu)

```
import numpy as np
# Implementacja rekurencyjnej sieci neuronowej do sumowania dwóch liczb 24-bitowych
def bin_add(a, b):
    """Funkcja dodająca dwie liczby binarne"""
    a = np.array([int(x) for x in a], dtype=int)
    b = np.array([int(x) for x in b], dtype=int)

    carry = 0
    result = []

    # Dodawanie bit po bicie
    for i in range(len(a)-1, -1, -1):
        total = a[i] + b[i] + carry
        result.append(total % 2)
        carry = total // 2

    if carry:
        result.append(carry)

    return ''.join(map(str, result[::-1]))

# Przykład dodawania dwóch liczb binarnych 24-bitowych
a = '11011011010110110101'
b = '101101101111010101110010'

print(bin_add(a, b))

1100100100101000011100111
```

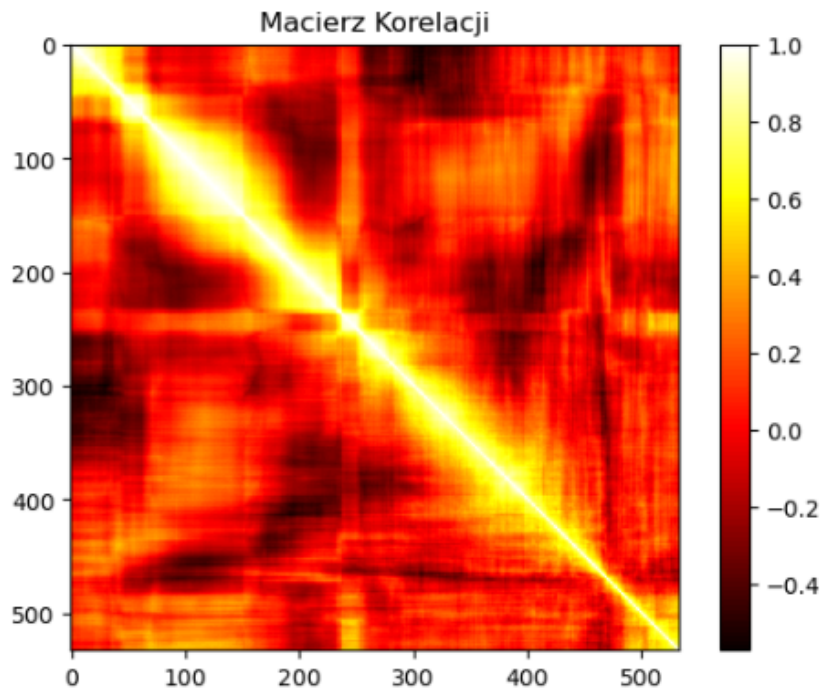
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import TruncatedSVD
from skimage import io
# Obliczanie korelacji obrazu z użyciem SVD
# Załaduj obraz
image_url = "6.webp" # obraz z wariantu 6
image = io.imread(image_url, as_gray=True)

# Obliczanie macierzy korelacji
corr_matrix = np.corrcoef(image)

# Rozkład SVD
svd = TruncatedSVD(n_components=2) # Przykład rozkładu na 2 komponenty
svd.fit(corr_matrix)

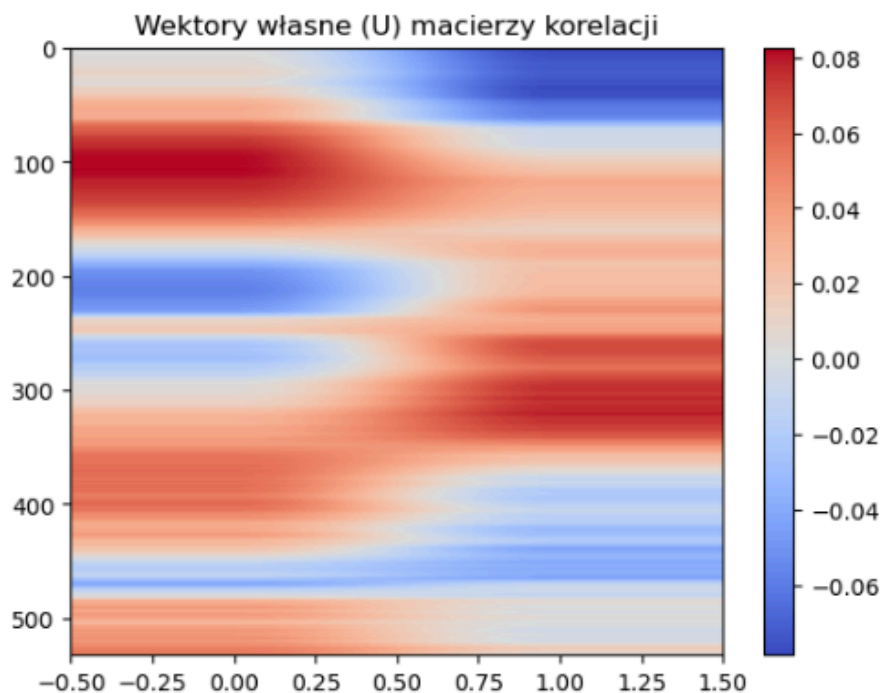
# Zbiór macierzy U, Σ, V
U = svd.components_.T # Wektory własne wierszy
V = svd.components_ # Wektory własne kolumn

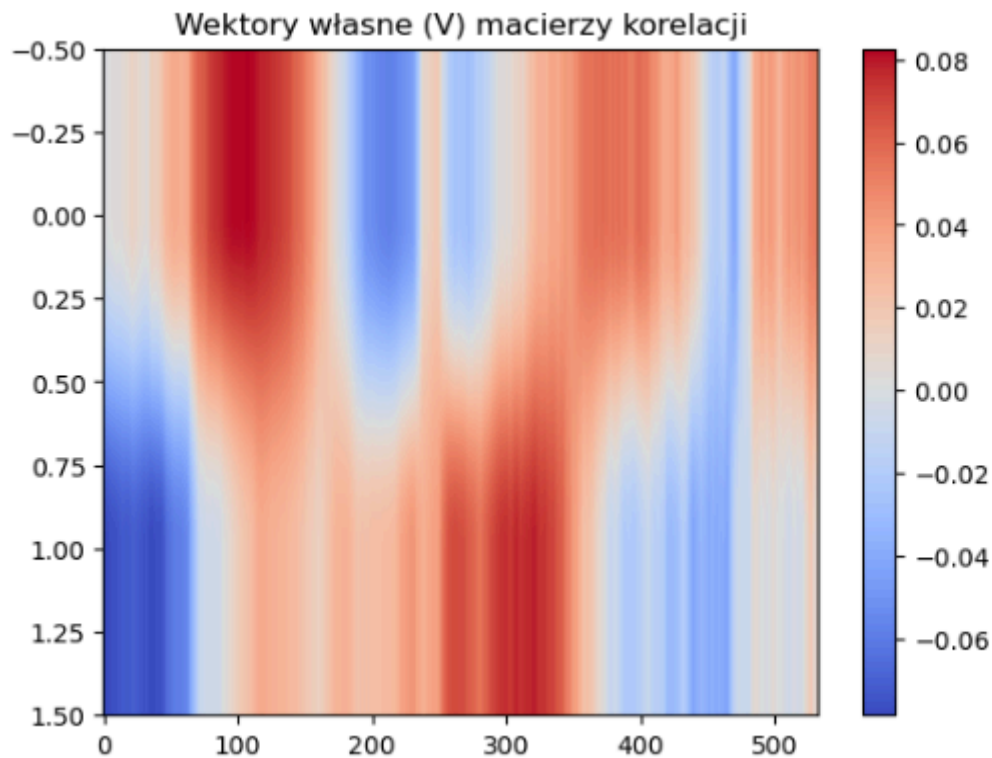
# Wizualizacja macierzy korelacji
plt.imshow(corr_matrix, cmap='hot', interpolation='nearest')
plt.colorbar()
plt.title("Macierz Korelacji")
plt.show()
```



```
[5]: #Wizualizacja wyników SVD i korelacji
# Wizualizacja pierwszej składowej SVD
plt.imshow(U, cmap='coolwarm', aspect='auto')
plt.colorbar()
plt.title("Wektory własne (U) macierzy korelacji")
plt.show()

# Wizualizacja drugiej składowej SVD
plt.imshow(V, cmap='coolwarm', aspect='auto')
plt.colorbar()
plt.title("Wektory własne (V) macierzy korelacji")
plt.show()
```





4. Wnioski

Zadanie ma na celu zapoznanie się z wykorzystaniem rozkładu SVD (Singular Value Decomposition) do analizy danych, w tym obliczania korelacji między danymi i rozkładu tych korelacji na różne składowe. Korzystając z SVD, możemy rozłożyć macierz korelacji na trzy macierze (U , Σ , V), które pozwalają na zrozumienie struktury danych, zarówno w kontekście wierszy, jak i kolumn. W zadaniu omawiamy także użycie macierzy korelacji, które są fundamentem analizy statystycznej oraz algorytmów takich jak PCA (Principal Component Analysis). Zastosowanie tej techniki w różnych dziedzinach, takich jak analiza obrazu czy sygnałów, pozwala na efektywne wyodrębnienie najważniejszych informacji i wykrywanie ukrytych wzorców.

Rozkład SVD, wykorzystywany do analizy macierzy korelacji, umożliwia głębsze zrozumienie struktury danych, co jest istotne w wielu dziedzinach, od analizy danych po rozpoznawanie wzorców w obrazach. Ponadto, zrozumienie i wykorzystanie SVD w kontekście analizy korelacji może pomóc w lepszym modelowaniu danych, wykrywaniu istotnych zależności oraz optymalizacji procesów w różnych aplikacjach naukowych i inżynierskich.