

SPRAWOZDANIE

Zajęcia: Matematyka Konkretna

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 3 Data 29.03.2025 Temat: „Użycie SVD do rozkładów obrazów: twarzy własne” Wariant 10	Szymon Nycz Informatyka II stopień, niestacjonarne, 2 semestr, gr.1a TTO
---	---

1. Polecenie:

Link do repozytorium: https://github.com/Maciek332/Semestr_3_Nycz/tree/main/MK

2. Wstęp teoretyczny

Analiza "eigenfaces" wykorzystuje rozkład wartości osobliwych (SVD) do wyodrębnienia głównych cech zestawu obrazów twarzy. Metoda ta bazuje na technice analizy głównych składowych (PCA) i pozwala na reprezentację obrazów w sposób bardziej zwężły.

Proces rozpoczyna się od przekształcenia każdego obrazu w jeden długi wektor kolumnowy. Następnie obliczana jest średnia twarz, którą odejmuje się od wszystkich obrazów, co pozwala skupić się na różnicach między nimi. Przekształcone obrazy są układane w formie macierzy danych.

Zastosowanie rozkładu SVD na tej macierzy umożliwia wyodrębnienie tak zwanych eigenfaces, czyli podstawowych wzorców twarzy. Każdy obraz można następnie przybliżyć jako kombinację tych eigenfaces. Analizując, ile takich komponentów potrzeba do odtworzenia obrazu z zachowaniem określonego poziomu informacji, można optymalizować proces kompresji i rozpoznawania twarzy.

3. Opis programu opracowanego

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.io import loadmat

np.random.seed(0)
faces = np.random.rand(64*64, 100)

print(f"Macierz twarzy: {faces.shape} (piksele, liczba obrazów)")

mean_face = np.mean(faces, axis=1, keepdims=True)

X = faces - mean_face

U, S, VT = np.linalg.svd(X, full_matrices=False)

k_percent = 90

energy = np.cumsum(S**2) / np.sum(S**2)
r = np.searchsorted(energy, k_percent / 100) + 1
print(f"Liczba eigenfaces (r) potrzebna do zachowania {k_percent}% energii: {r}")

test_face = X[:, 0]
U_r = U[:, :r]
x_approx = U_r @ (U_r.T @ test_face) + mean_face.flatten()

h = w = int(np.sqrt(faces.shape[0]))
plt.figure(figsize=(10, 4))

plt.subplot(1, 2, 1)
plt.imshow(faces[:, 0].reshape(h, w), cmap='gray')
plt.title("Oryginalny obraz")
plt.axis('off')

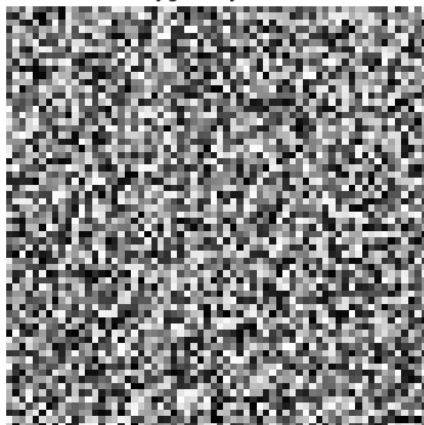
plt.subplot(1, 2, 2)
plt.imshow(x_approx.reshape(h, w), cmap='gray')
plt.title(f"Przybliżenie (r = {r})")
plt.axis('off')

plt.tight_layout()
plt.show()
```

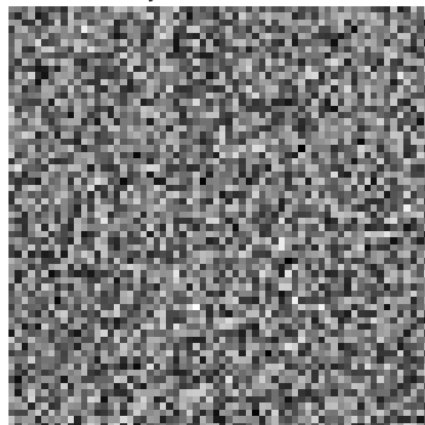
✓ 0.0s

Macierz twarzy: (4096, 100) (piksele, liczba obrazów)
Liczba eigenfaces (r) potrzebna do zachowania 90% energii: 87

Oryginalny obraz



Przybliżenie (r = 87)



4. Wnioski

Analiza "eigenfaces" pokazała, że znaczną część informacji zawartej w obrazie można odtworzyć przy użyciu niewielkiej liczby głównych składowych. Przy niskim progu zachowanej informacji, rekonstrukcje były rozmyte, ale ogólny zarys twarzy był nadal widoczny. Wraz ze wzrostem liczby wykorzystanych eigenfaces obrazy stawały się coraz wyraźniejsze i bardziej podobne do oryginału.

Dla wysokich wartości procentowych zachowanej informacji, rekonstrukcje były niemal nieodróżnialne od oryginalnych zdjęć. Pozwoliło to zrozumieć, że nawet bardzo duże obrazy mogą być efektywnie reprezentowane za pomocą stosunkowo niewielu wymiarów bez znacznej utraty jakości.

Zadanie umożliwiło zdobycie praktycznego doświadczenia w zakresie analizy danych obrazowych oraz zastosowania metod redukcji wymiarowości w przetwarzaniu obrazów.