Przetwarzanie i analiza obrazów 16.03.2019

Projekt nr 1

Paweł Lorek

Termin oddania: 5.05.2019

Do oddania: Raport w postaci pliku .pdf oraz plik(i) w Python (ver 3).

Uwaga: Poniżej znajduje się opis co powinno znaleźć sie w projekcie. Twój projekt powinien te wszystkie elementy zawierać, ale nie musi się tylko i wyłącznie do nich sprowadzać.

Część A: Rozkład SVD i kompresja obrazów (10pkt)

Opis projektu

Wykonaj dekompozycje SVD dla obrazka
http://www.math.uni.wroc.pl/~lorek/image_analysis/images/baboon.bmp
oraz dla min. 3 przez siebie wybranych czarno-bialych obrazków (najlepiej tego samego rozmiaru bitmapy *.bmp*) Postaraj się, aby były tam obrazy zarówno skomplikowane (prawie każdy mały obszar ma dużo szczegółów) jak i "proste" (np. jest dużo obszarów o jednakowym lub mało się zmieniajacym kolorze)
Każdy obraz f przedstaw jako

$$f = \sum_{i=1}^{r} \lambda_i^{\frac{1}{2}} \mathbf{u}_i \mathbf{v}_i$$

Nazwijmy $\lambda_i^{\frac{1}{2}} \mathbf{u}_i \mathbf{v}_i$ obrazem własnym (eigenimage) odpowiadającym wartości własnej λ_i. Posortujmy wartości własne w taki sposób, by $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r$. Oznaczmy $f_k = \sum_{i=1}^{k} \lambda_i^{\frac{1}{2}} \mathbf{u}_i \mathbf{v}_i$. Dla każdego obrazka przedstaw (jako obrazek) kilka pierwszych obrazów własnych oraz f_k dla kilku wybranych k (zwiększające sie o stałą liczbę).
Przez kompresję obrazu f metodą rozkładu SVD nazywamy obraz f_k z odpowiednio dobranym parametrem k. Wypracuj jakiś sposób wyznaczania tego parametru (np. k ma być taką liczbą, by $\lambda_{k+1} + \ldots + \lambda_r$ stanowiły maksymalnie jakiś ustalony procent całej sumy $\sum_{i=1}^{r} \lambda_i$, procent ten może być wybrany eksperymentalnie na jednym obrazku).
Następnie wylicz w tak ustalony sposób k dla każdego obrazka. Czy wielkość k zależy od stopnia skomplikowania obrazka?

Policz też ile pamięci zajmuje tak skompresowany obraz. Porównaj wynik do obrazów (standardowych) *.jpg* otrzymanych na `chaosie` za pomocą komendy:
`chaos:~$ convert obrazek.bmp obrazek.jpg`

Zakładamy, że wszystkie obrazy są 8-bitowe (= 1 bajt) (skala szarości 0-255). Zatem np. jeden wektor $\mathbf{u_i}$ jest wektorem rozmiaru 256, i każdy element to 1 bajt, czyli $\mathbf{u}_i$ jest

zajmuje 256 bajtów = 0.25 Kb (1 Kb = 1024 bajty). Natomiast dla przykładu: obrazek 256×256 trzymany w postaci 8-bitowej bitmapy zajmuje: $256 \cdot 256 \cdot 8$ bitów = 524288 bitów = 524288/ (1024· 8)= 64 Kb.
(UWAGA: rozmiar pliku *.bmp* może być większy z kilku powodow: np. mimo iż obraz jest czarnobiały, to każdy kanal r,g,b jest i tak pamietany osobno, wtedy rozmiar to będzie $3 \cdot 64$Kb, albo może być 16, 24 lub 32-bitowy, trzymane też mogą być informacje o rozdzielczości).

Scieżka do zdjęcia powinna być parametrem programu (użyj `argparse`).

Część B: PCA + Klasyfikacja obrazów (10pkt)

Baza twarzy `Olivetti faces` składa sie z 400 obrazów (w skali szarości) rozmiaru 64×64. Są to zdjęcia 40 osób (po 10 zdjęć na osobę).
Bazę wczytujemy w następujący sposób:

```
from sklearn.datasets import fetch_olivetti_faces
faces = fetch_olivetti_faces()
```

(zob.: `faces.keys(), faces.images.shape, faces.data.shape, faces.target.shape`)

Do dalszej analizy wybierz tylko zdjęcia 10 (losowo wybranych) osób. Otrzymany zbiór podziel losowo na dwie części

- Zbiór *treningowy*: po 7 zdjęć każdej z 10 osób.
- Zbiór *testowy*: pozostałe zdjęcia (tj. po 3 dla każdej z osób)

Zadania

- Wykonaj na zbiorze treningowym PCA redukując wymiar 64^2 do różnych wartości d (w tym $d = 0$, tzn. de facto bez PCA)
- Przekształć punkty ze zbioru testowego za pomocą otrzymanej macierzy przekształcenia z poprzedniego punktu.
- Wykonaj klasyfikację kNN tak przekształconych obrazów z różnymi wartościami parametru k
- Podaj dla testowanych par d i k wynik klasifikacji (classification rate). Jakie wartości d i k dały najlepsze wyniki?

Dodatkowo, dla kilku par d i k (w tym dla tej, która dała najlepsze wyniki dla kNN) zamiast kNN zastosuj: `GaussianNB, RandomForestClassifier, SVC`, ich użycie jest takie samo jak kNN (możesz używać ich domyślnych parametrów), przykładowe użycie jest np. na stronie:
https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html