

PCA Selfie

10 listopada 2017

Grzegorz Borkowski

1. WSTĘP

Zadanie zostało zaimplementowane w Pythonie 3.6.1
Do raportu został dołączony Jupyter Notebook z kodem
źródłowym programu.

2. ZADANIE

2.1 Treść zadania

Robimy 15+ (im więcej tym lepiej) różnych selfie (z
możliwie tym samym tłem i oświetleniem, niekoniecznie
z tym samym wyrazem twarzy)

Konwertujemy je do czerni i bieli oraz zmniejszamy ich
rozdzielczość. Traktujemy je jako zbiór $n \cdot (15+)$ elemen-
tów należących do n klas. Dokonujemy zamiany zdjęć na
wektory i przeprowadzamy na takim zbiorze PCA (zwykle,
nie kernel).

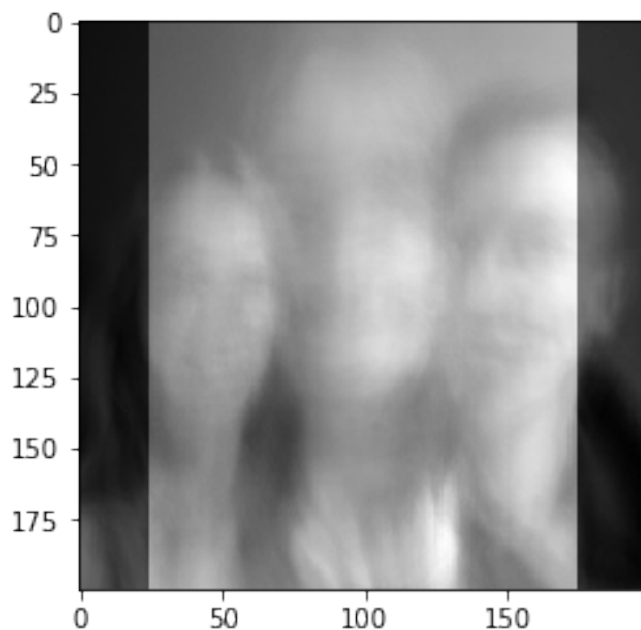
Jak wygląda "średnia twarz?". Jak wyglądają kolejne
"principal components" (są bardzo długimi wektorami, ale
możemy je przekonwertować ponownie na zdjęcia i tak
zaprezentować)? Zauważmy też, że tylko niewielka część
nowych wektorów bazowych ma istotny udział w tłumacze-
niu pierwotnej wariancji zbioru (explained variance ratio).
Jak wyglądają zrekonstruowane twarze, jeżeli obetniemy
przestrzeń do 5, 15 i 50 najważniejszych? Na koniec,
ograniczmy przestrzeń do 2 najważniejszych wymiarów i
zrzutujemy elementy zbioru na płaszczyznę 2D (kolorując
punkty w zależności od klasy do której przynależą). Czy
są łatwo separowalne?

2.2 Zbiory danych

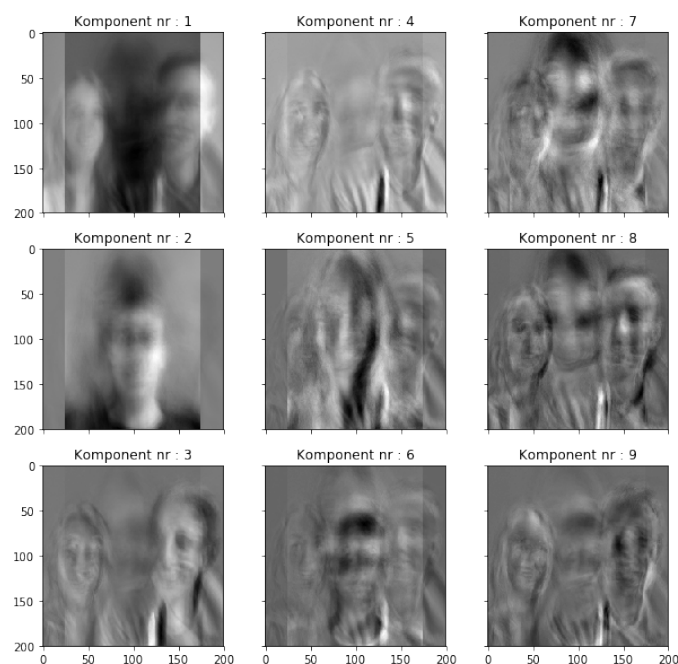
Zbiór danych składa się z 216 elementów, podzielonych na
trzy klasy (klasa "Ania", klasa "Grzegorz", klasa "Ania
i Grzegorz"). Klasy te mają odpowiednio po 81, 82 i
53 elementy. Zdjęcia te zostały zmniejszone do rozmiaru
200x200 pikseli każde i zmieniono im kolor na odcienie
szarości.

2.3 Eksperyment

Wykonałem Principal Component Analysis na całym
zbiorze zdjęć. Wyliczyłem średnie zdjęcie. Wybrałem 9
pierwszych wektorów własnych i stworzyłem z nich zdję-
cia. Następnie aby zbadać stopnie kompresji, wybierałem
liczbę pierwszych wektorów kolejno ze zbioru 1, 2, 3, 4, 5,
10, 15, 25, 50, 100, 150, 200 i rekonstruowałem 9 zdjęć,
po 3 z każdej klasy wykorzystując wybrane wektory. Na
końcu wybrałem dwa pierwsze komponenty i zrzutowałem
zdjęcia tworząc wykres dwuwymiarowy elementów.



Rysunek 1. "Średnie" zdjęcie



Rysunek 2. Zdjęcia reprezentujące wektory własne

Rekonstrukcja z 1 wektorow



Rekonstrukcja z 3 wektorow



Rysunek 3. Rekonstrukcja wykorzystująca jeden wektor

Rysunek 5. Rekonstrukcja wykorzystująca trzy wektory

Rekonstrukcja z 2 wektorow



Rekonstrukcja z 4 wektorow



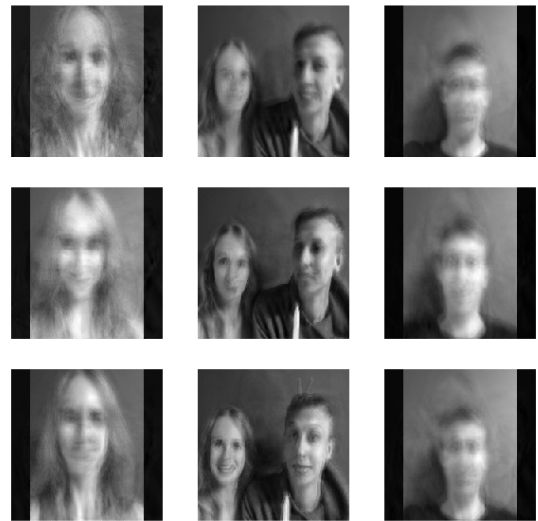
Rysunek 4. Rekonstrukcja wykorzystująca dwa wektory

Rysunek 6. Rekonstrukcja wykorzystująca cztery wektory

Rekonstrukcja z 5 wektorów



Rekonstrukcja z 25 wektorów



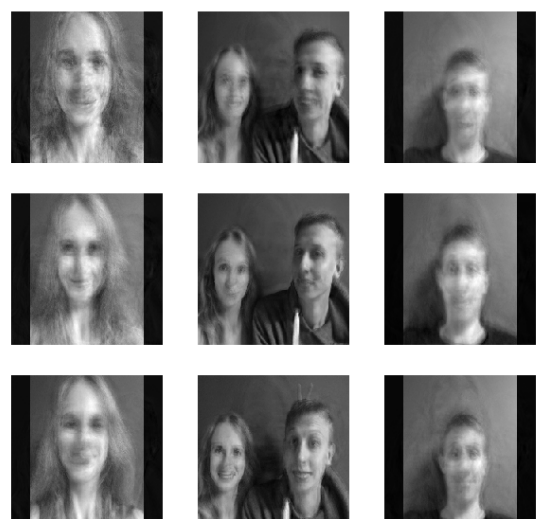
Rysunek 7. Rekonstrukcja wykorzystująca pięć wektorów

Rysunek 9. Rekonstrukcja wykorzystująca dwadzieścia
piec wektory

Rekonstrukcja z 10 wektorów



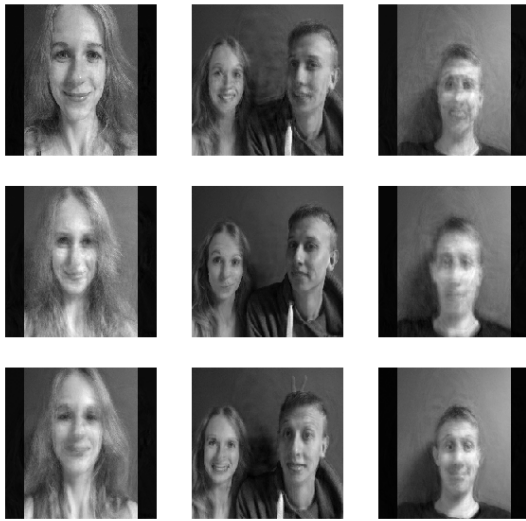
Rekonstrukcja z 50 wektorów



Rysunek 8. Rekonstrukcja wykorzystująca dziesięć wekt-
rów

Rysunek 10. Rekonstrukcja wykorzystująca piędziesiąt
wektory

Rekonstrukcja z 100 wektorów



Rekonstrukcja z 200 wektorów



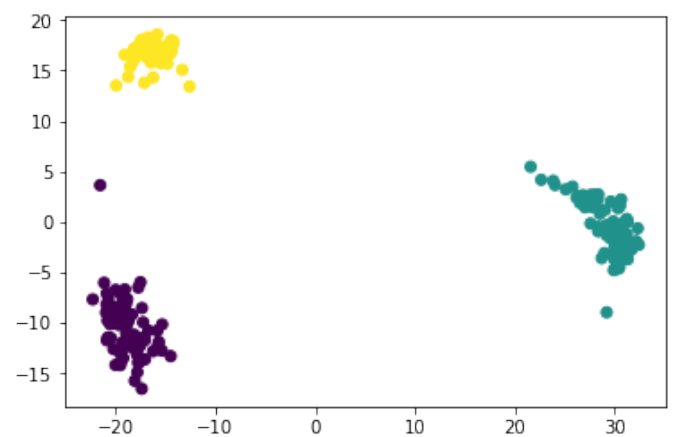
Rysunek 11. Rekonstrukcja wykorzystująca sto wektorów

Rysunek 13. Rekonstrukcja wykorzystująca dwieście wektorów

Rekonstrukcja z 150 wektorów



Rysunek 12. Rekonstrukcja wykorzystująca sto pięćdziesiąt wektorów



Rysunek 14. Rzut elementów zbioru na płaszczyznę 2D

3. WNIOSKI

- (1) PCA w przypadku zdjęć o tej samej rozdzielczości, oświetleniu, kolorach, tło jest w stanie bardzo dobrze redukować wymiarowość. Przy pomocy tylko jednego wektora własnego, jesteśmy w stanie odtworzyć esencję zdjęcia, a przy pomocy dwustu wektorów kompresja jest niemal niezauważalna.
- (2) Istnieją wektory własne reprezentujące elementy z konkretnej klasy (Rysunek 2 - zdjęcie reprezentujące wektory własne - komponent nr 2) lub pewną formę sumy lub złożenia elementów z różnych klas (komponenty 5,7,8,9)
- (3) Rzut zbioru wykorzystując tylko dwa wektory własne w przypadku selfie jest wystarczający, aby poprawnie zaklasyfikować obraz do klasy (widać na wykresie wyraźne klastry - klasyfikator k najbliższych sąsiadów poradziłby sobie w tym przypadku doskonale)