

Opis PCA projekt

Jakub Banaśkiewicz

1 Wstęp

W projekcie zostały zaimplementowane dwa algorytmy kompresji. Głównie opierają się one na idei podziału zdjęcia na mniejsze i wykonywaniu na nim PCA.

2 Algorytm I

Mając dane zdjęcie które jest tensorem $n \times m \times 3$ dzielimy obrazek na ciąg X tensorów wymiarów $h \times w \times 3$ gdzie $n|h$, $m|w$ i traktując każdy z elementów jako wektor liczymy macierz kowariancji X . Niech $d \in \{1, \dots, 3 * h * w\}$, by zakodować obrazek od każdego elementu X odejmujemy ich średnia a potem każdy z nich przekształcamy przez odwzorowanie P^T gdzie $P = [p_1, \dots, p_d]$ gdzie p_1, \dots, p_d jest ciągiem wektorów własnych macierzy kowariancji, które odpowiadają największym wartościom własnym. Kodem jest ciąg wszystkich takich wektorów utworzonych z X . By zdekodować zakodowany obrazek dla pewnego $d \in \{1, \dots, 3 * h * w\}$ należy każdy element przekształcić za pomocą P i dodać do niego średnia. Potem należy złożyć obraz z powrotem wykorzystując otrzymane elementy

3 Algorytm II

Drugi algorytm polega na zastosowaniu algorytmu pierwszego do pewnego podziału wyjściowego zdjęcia i potem złożenia go z powrotem. W przypadku podziału na tylko na jeden element algorytm ten redukuje się do algorytmu pierwszego.

4 Ile miejsca można zaoszczędzić

W pierwszym algorytmie należy zapamiętać $w * h * 3 * (d+1)$ liczb przeznaczonych na wektory własne i średnia oraz $(n/h) * (m/w) * d$ liczb na kodowane X .

W drugim algorytmie należy tą liczbę wartości własnych przemnożyć przez liczbę na jaką podzieliśmy obrazek.

5 Eksperymenty

Dla testowanego zdjęcia kompresja JPG zajmuje mniej niż 4% miejsca. Uzyskała ona wynik 95% na metryce SSIM. Kompresje z PCA dla, której udało mi się uzyskać taki wynik zajmowały co najmniej 24% miejsca oryginalnego obrazka.