Jastrzębski Piotr 50764 ISM

Próbkowanie i zmiana rozmiaru obrazu, kwantyzacja oraz dithering

2.1.1

Zbadać wizualną skuteczność działania (oddalenie przybliżenie zdjęcia) w zależności od stopnia zmiany rozmiaru (sprawdzić dla więcej niż dwóch rozmiarów przekształceń)

Skalowanie obrazów kolejno 5%, 50%, 120%, 150%.

a) obraz 0008.png

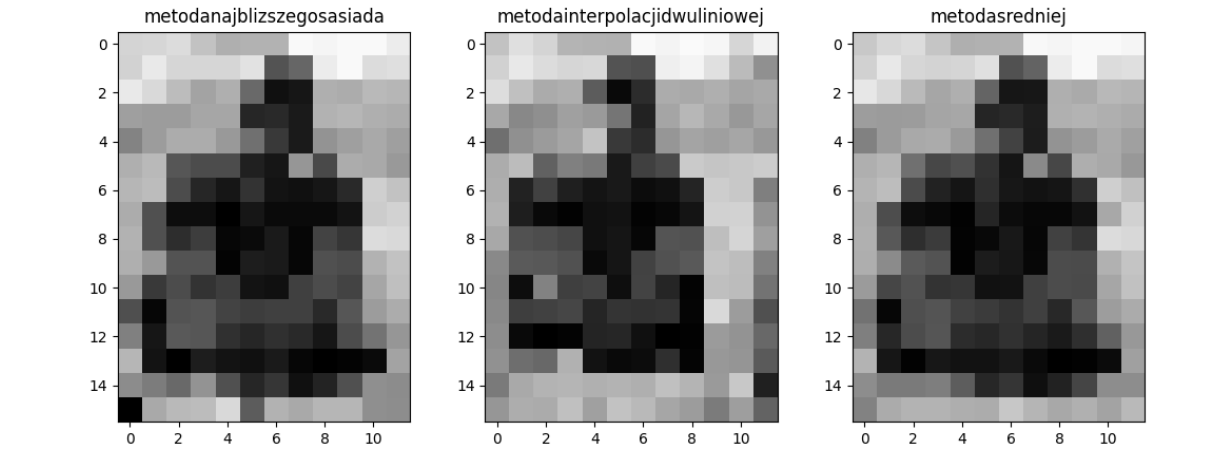


Figure 1 5% rozmiaru

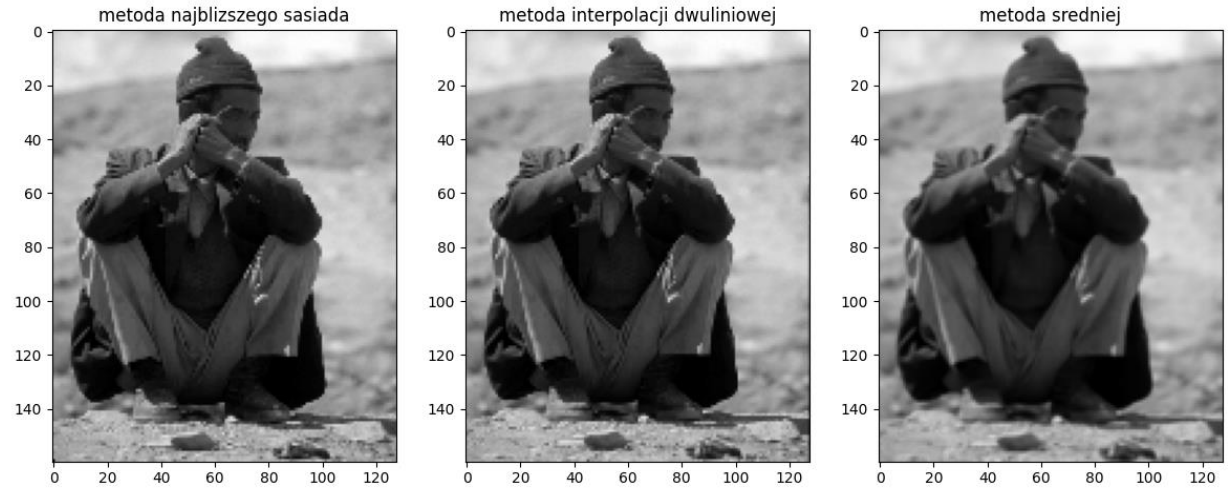


Figure 2 50% rozmiaru

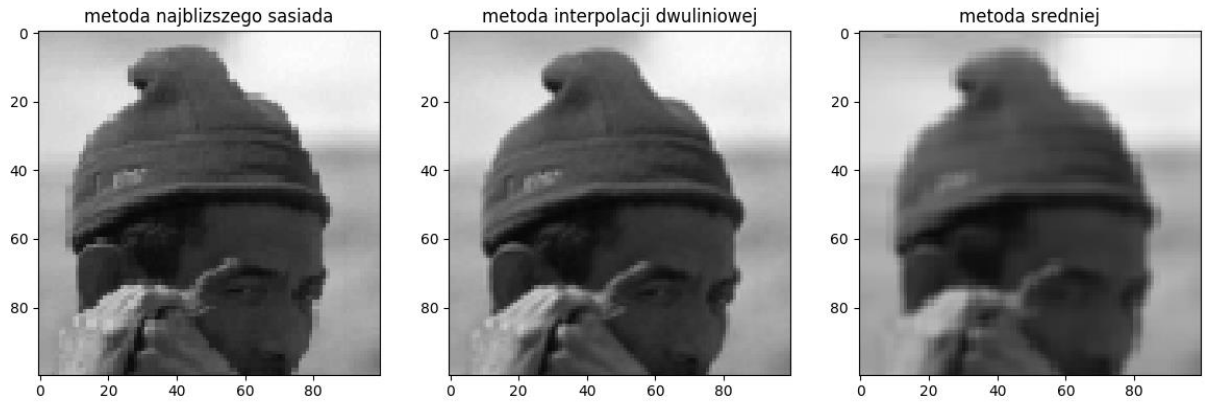


Figure 3 120% rozmiaru

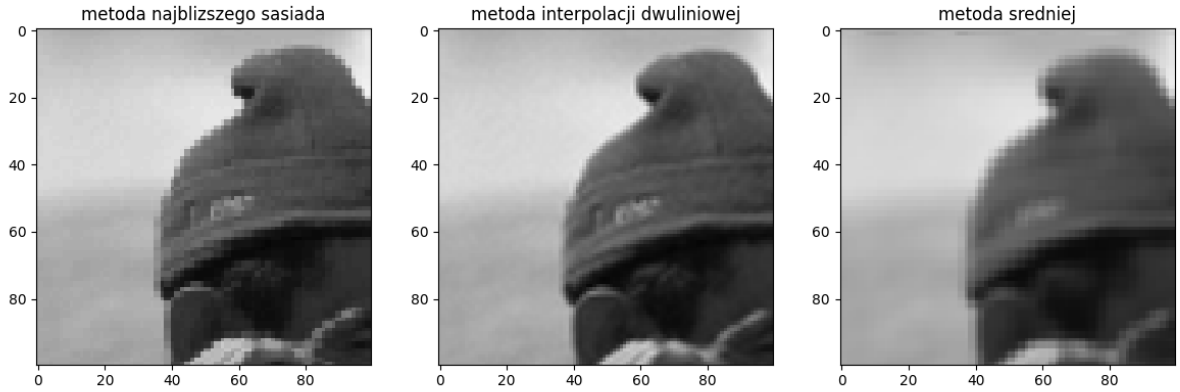


Figure 4 150% rozmiaru

b) 0009.png

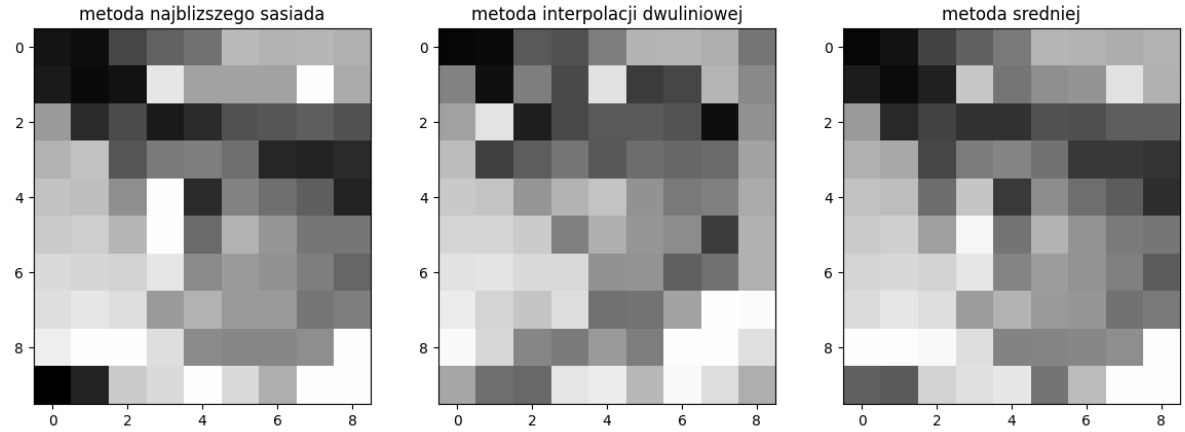


Figure 5 5% rozmiaru

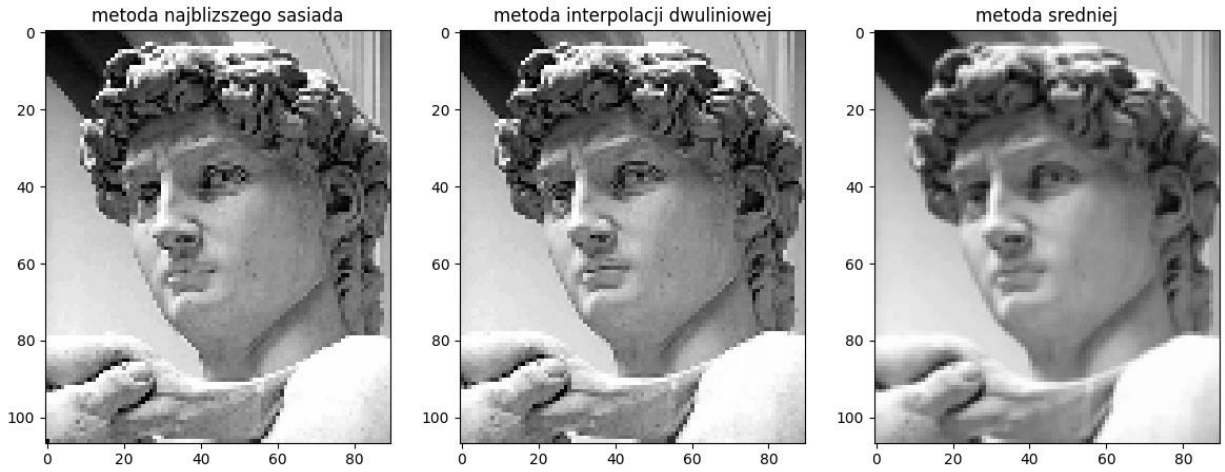


Figure 6 50% rozmiaru

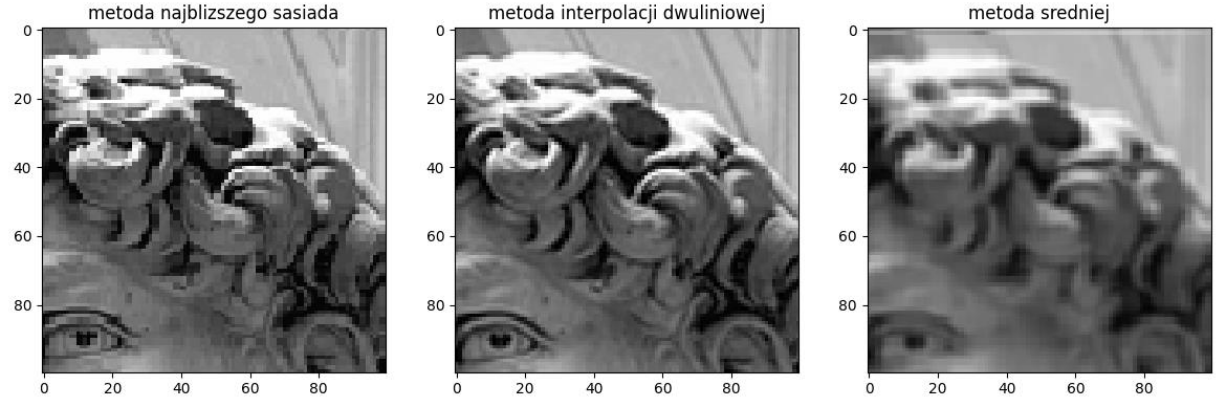


Figure 7 120% rozmiaru

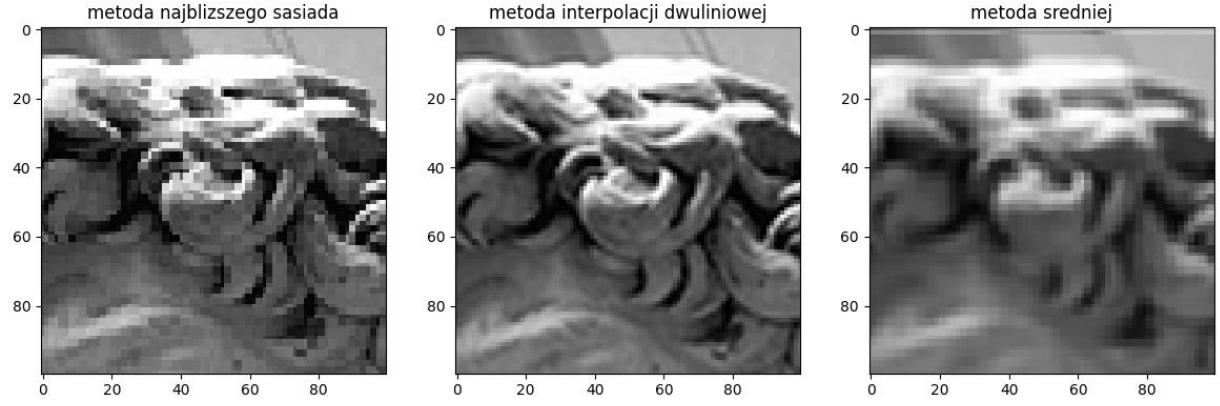


Figure 8 150% rozmiaru

c) obraz o wymiarach 3000x4000 – rozmiary kolejno: 100%, 5%, 5%, 50%, 120%, 150%

Figure 9 rozmiar 100%

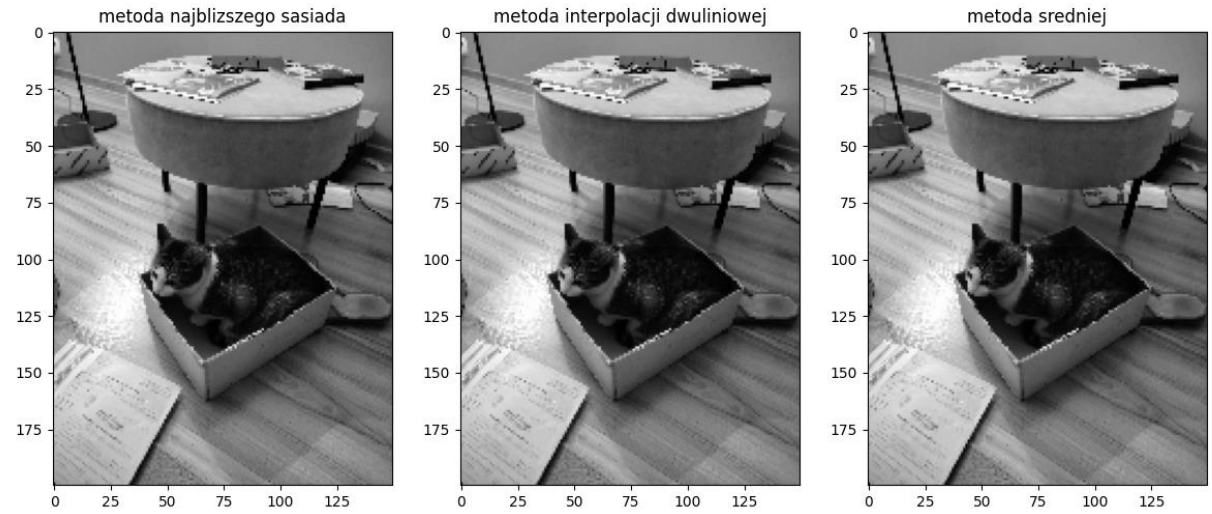


Figure 10 rozmiar 5%

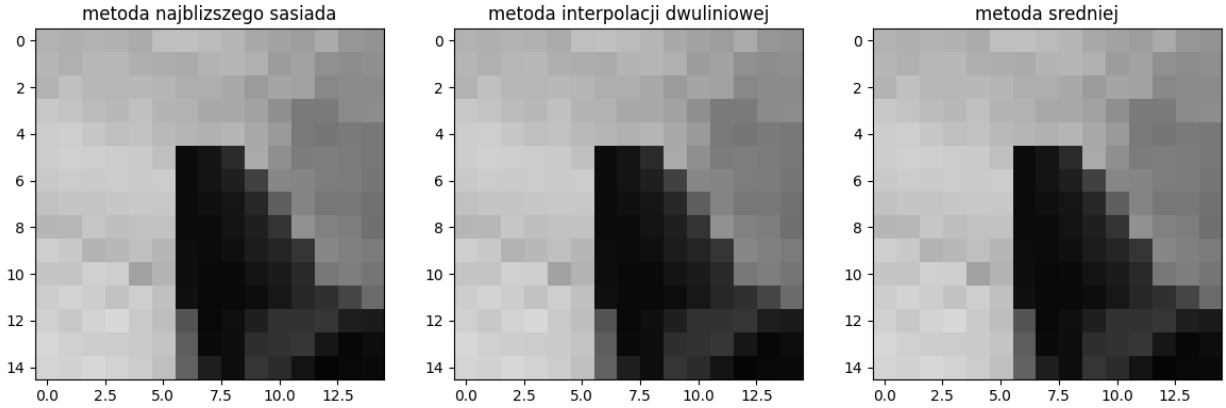


Figure 11 rozmiar 5%

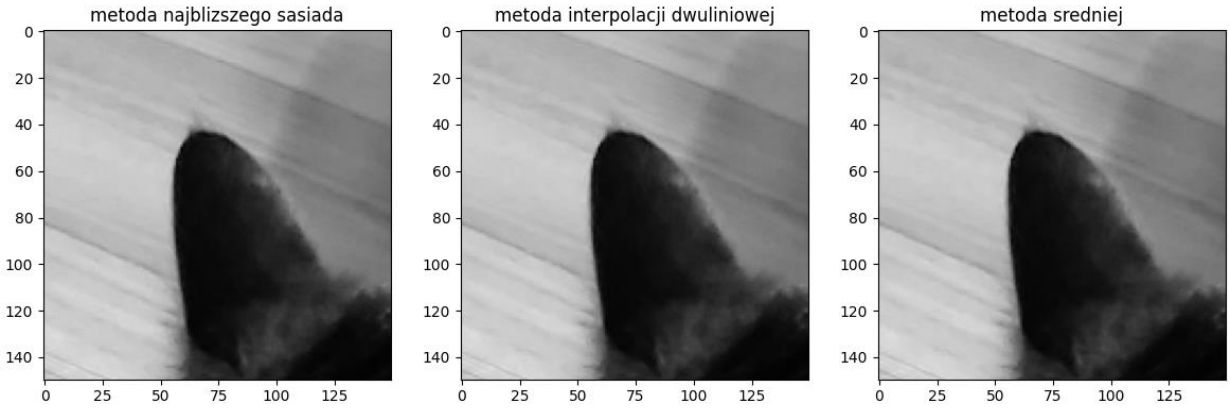


Figure 12 rozmiar 50%

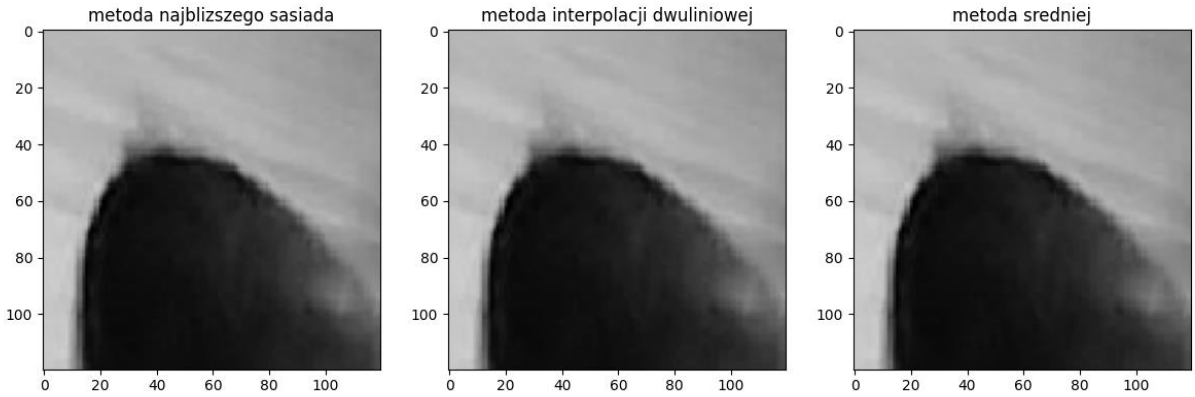


Figure 13 rozmiar 120%

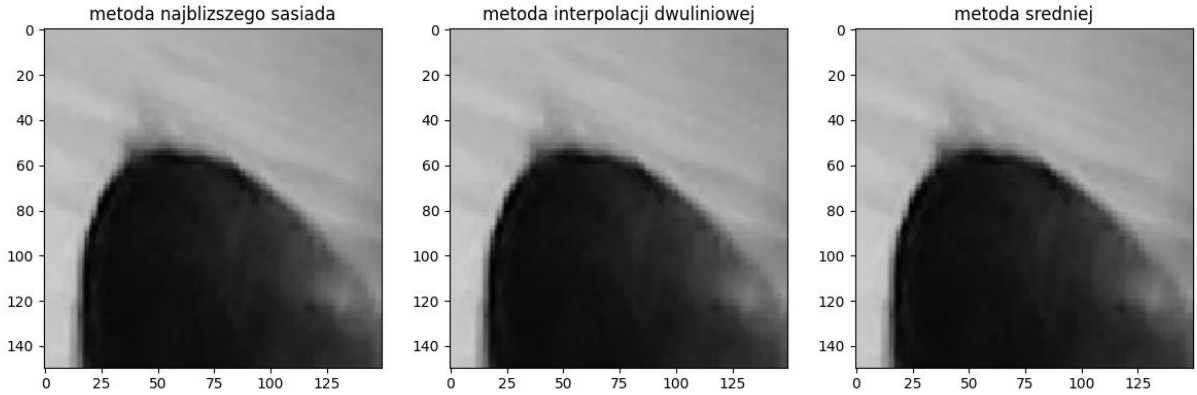


Figure 14 rozmiar 150%

Wnioski:

Przy drastycznej redukcji rozmiarów (w tym przypadku do 5% wielkości), metoda interpolacji okazała się subiektywnie najgorsza, gdzie pozostałe dwie zachowały ogólny balans kolorów.

Przy 50% rozmiaru zaś, metoda najbliższego sąsiada tworzy "ząbki" i obraz wygląda niełądnie.Wyraźnie lepszą jakość przedstawia metoda interpolacji, która jednak wciąż wygląda niedokładnie. Metoda średnich wygląda najwierniej oryginałom, ale jest wyraźnie rozmyta.

Przy powiększaniu rozmiarów, metoda interpolacji okazała się bezkonkurencyjna, prezentując wyraźnie lepszą dokładność i ostrość.

Przy ogromnej rozdzielczości początkowej obrazu (3000x4000), wybór algorytmu wydaje się nie mieć znaczenia, obrazy wyglądają niemal identycznie, a już przy 50% rozmiaru nie widzę żadnych różnic, co utrzymuje się również przy powiększaniu obrazu.

2.1.2

Zbadać wpływ działania algorytmów na wykrywanie krawędzi w obrazie (przykładowo wykorzystać do tego funkcję cv2.Canny). Sprawdzić czy są parametry dla których niektóre krawędzie przestaną być wykrywane w sposób prawidłowy.

a) 0008.png, krawędzie wyglądają tak samo dla algorytmów zmieniających rozmiar, parametr ksize = 5,

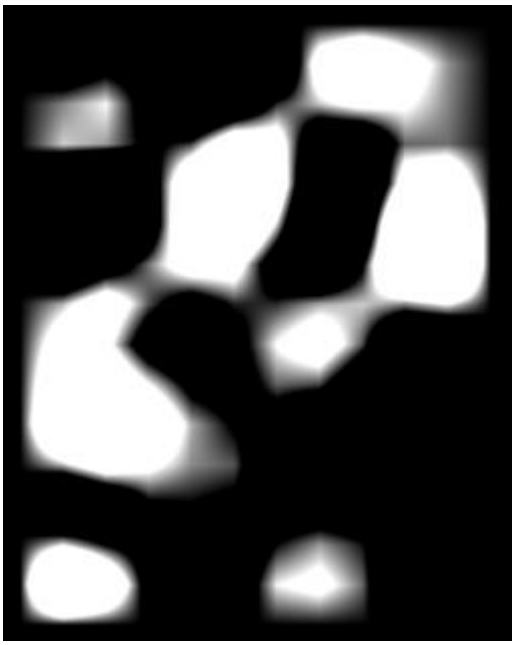


Figure 15 krawedzie dla 5% wyskalowane za pomocą cv2.resize(sobelxy, (256, 320))

Figure 16 krawedzie dla 50%

Figure 17 krawedzie dla 120%

Figure 18 krawedzie dla 150%

b) Zmiana parametru funkcji cv2.Canny

Zmiana parametru ksize dla obrazu o rozmiarze 50%, gdzie krawędzie były już bardzo dobrze widoczne (parametr ksize zawiera się w <1,31>, wartości całkowite, nieparzyste:

Figure 19 ksize = 7

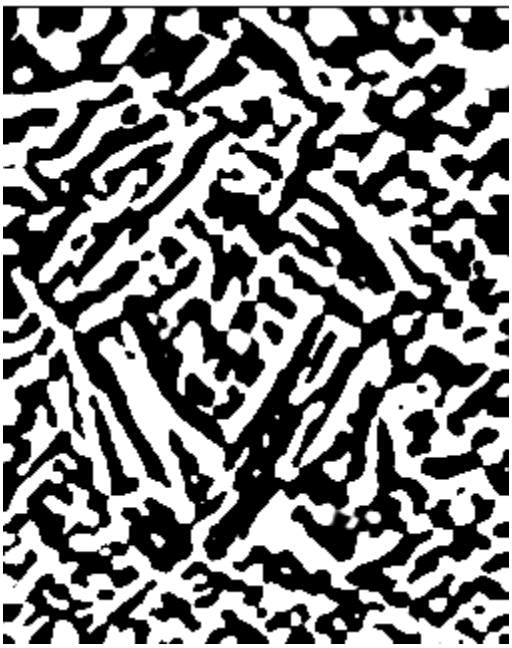


Figure 20 ksize = 9

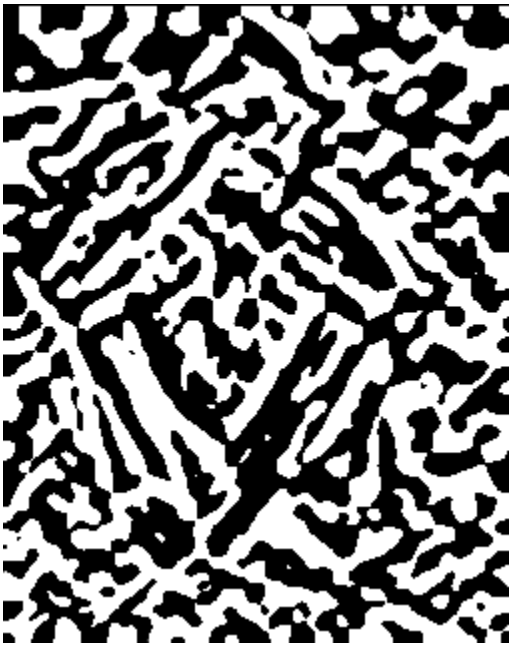


Figure 21 ksize=11

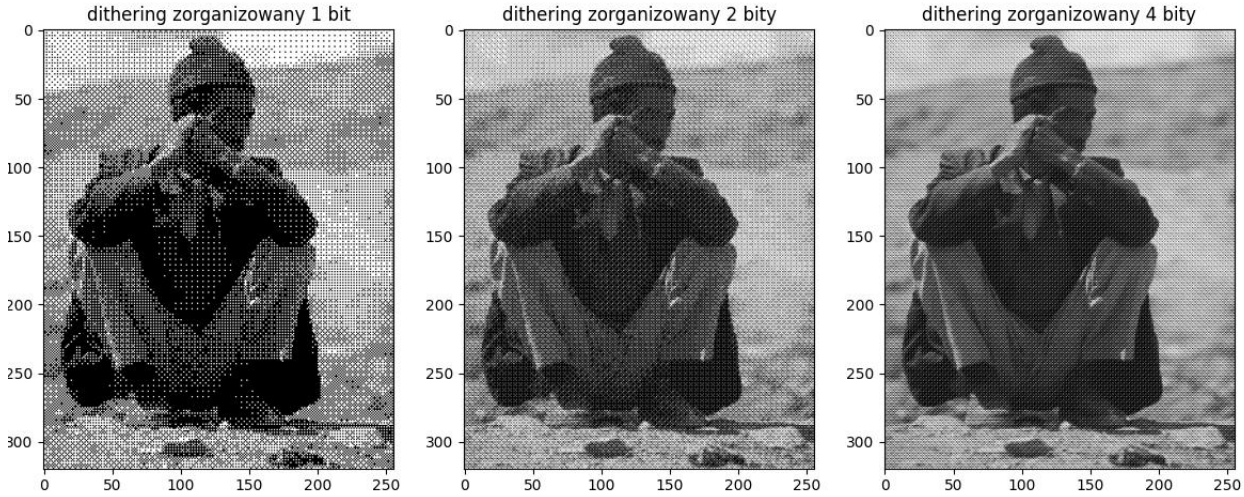
Wynik nie zmienia się znacznie, a co dla ksize < 5?

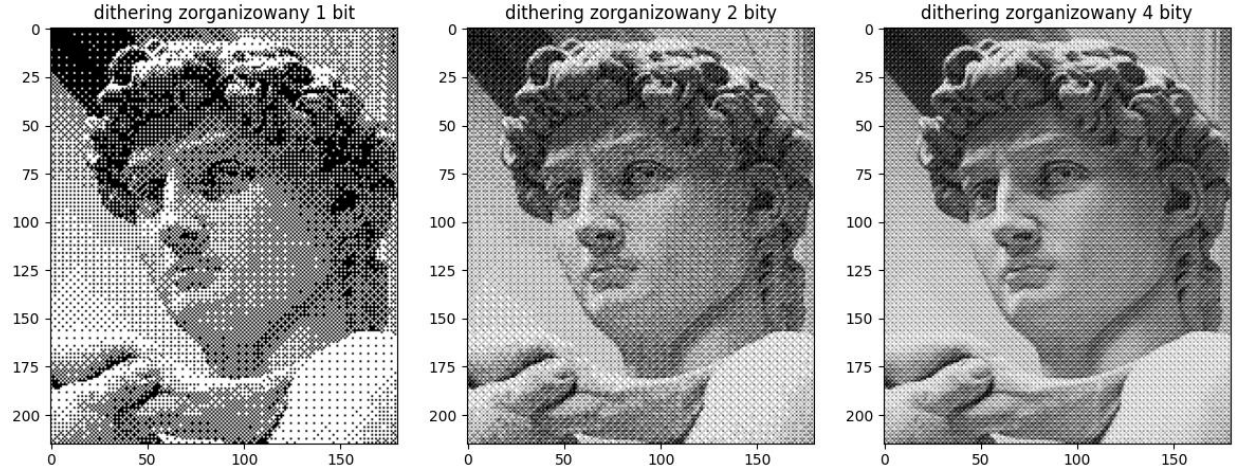
Figure 22 ksize = 3

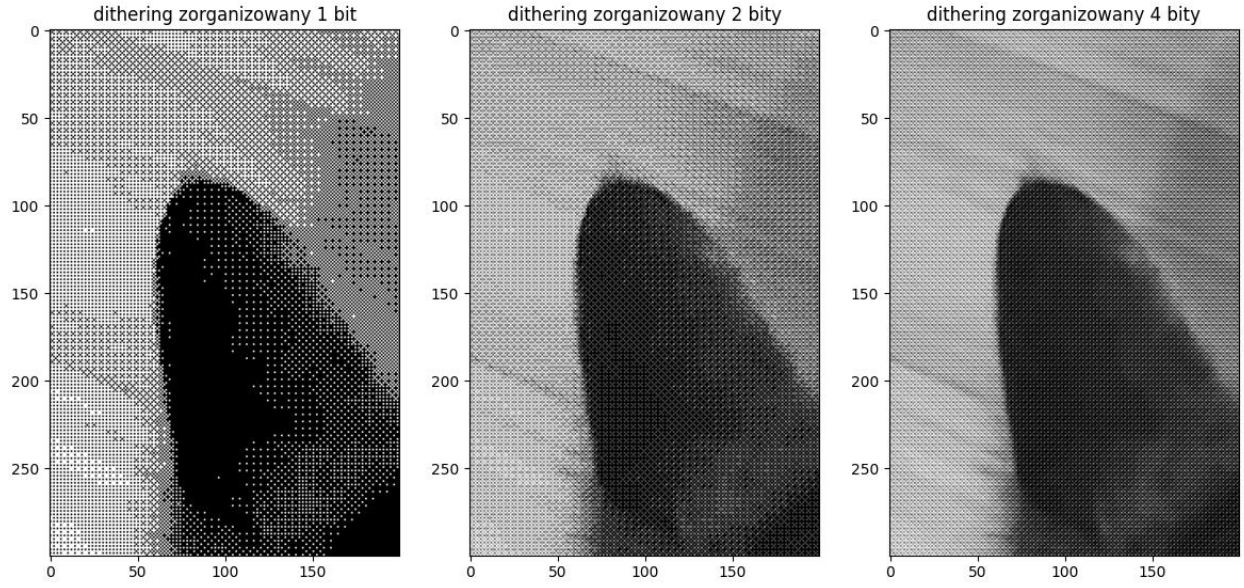
Figure 19 ksize = 1

2.2 Przebadać ich skuteczność działania na załączonych próbkach obrazów w skali odcieni szarości (wybrać kilka z nich). Ocenić efekty. Sprawdzić różnice dla obrazów w skali odcieni szarości zapisanych na 1,2,4 - bitach.

a) dithering zorganizowany


dithering zorganizowany 1 bit
dithering zorganizowany 2 bity
dithering zorganizowany 4 bity


dithering zorganizowany 1 bit
dithering zorganizowany 2 bity
dithering zorganizowany 4 bity


dithering zorganizowany 1 bit
dithering zorganizowany 2 bity
dithering zorganizowany 4 bity

b) dithering floyda-steinberga

Wnioski:

Dla 1 bita, oba algorytmy ditheringu wyglądają widocznie niskiej jakości. Przy 2 bitach jakość obrazów drastycznie rośnie, gdzie dla obrazu wysokiego kontrastu, dithering Floyda-Steinberga prezentuje się znacznie lepiej, widać dużo mniej artefaktów. Dla 4 bitów, dithering zorganizowany wypada znacznie gorzej, wciąż widać regularną siatkę, gdzie drugi algorytm wygląda prawie jak oryginał.