# Systemy multimedialne

Daniel Brzezicki (bd46477)

Lab – 04.05.2022

## Zadanie 1

Do zaimplementowania (0.7 pkt):

Algorytm realizujący nasz uproszczony algorytm JPEG dający wybór pomiędzy 4 kombinacjami poniższych kombinacji:

a. Redukcję chrominancji - wybór pomiędzy 4:4:4 a 4:2:2
b. Wybór tablicy kwantyzującej lub pominięcie jej (albo zastąpienie jej tablicą jedynek)

```python
def chromaSubsample(data, params):
    y,x = data.shape

    result=np.empty((y, int(x/2)))
    if params=="4:2:2":
        for _y in range(0,y):
            for _x in range(0,x,2):
                result[_y][int(_x/2)]=data[_y][_x]
    else:
        return data

    return result
```

```python
def chromaDesubsample(data, params):
    y,x = data.shape

    result=np.empty((y, x*2))
    if params=="4:2:2":
        for _y in range(0,y):
            for _x in range(0,x):
                result[_y][_x*2]=data[_y][_x]
                result[_y][_x*2+1]=data[_y][_x]
    else:
        return data

    return result
```

```python
def compress(data, params, tab):
    cs=chromaSubsample(data,params)
    cs = cs.astype(int)-128
    cs_y,cs_x = cs.shape
    result=np.zeros(cs_y*cs_x)

    dct = dct2(cs)
    dct_y,dct_x = dct.shape

    idx=0
    for _y in range(0,dct_y,8):
        for _x in range(0,dct_x,8):
            result[idx:idx+64]=np.round(zigzag(dct[_y:_y+8,_x:_x+8])/tab.flatten())
            idx+=64
    return result
```

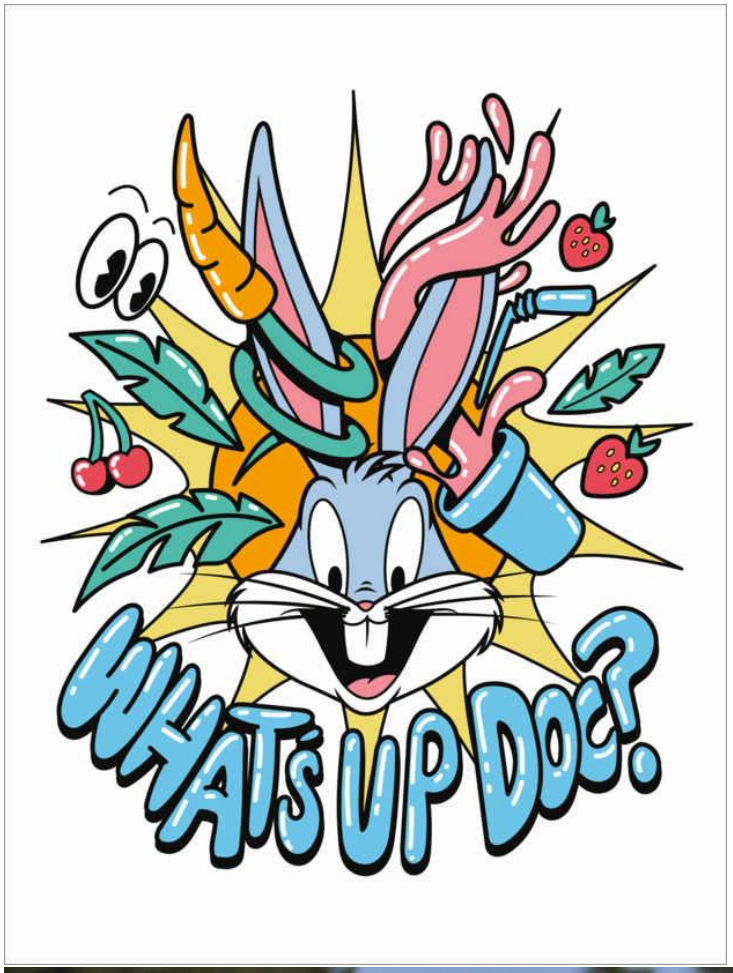
```python
def decompress(data,params,tab):
    result = initData(params,data.shape[0])

    y,x=result.shape
    for idx, i in enumerate(range(0,data.shape[0],64)):
        dequantized=data[i:i+64]*tab.flatten()
        _x=(idx*8)%x
        _y=int((idx*8)/x)*8
        result[_y:_y+8, _x:_x+8]=zigzag(dequantized)

    udct=idct2(result)+128
    result= chromaDesubsample(np.clip(udct,0,255).astype(np.uint8),params)
    return result
```

# Zadanie 2

Sprawozdanie/raport z działania programu (0.3 pkt):

1. Wybrać kilka (więcej niż 2) różnych dużych zdjęć i przeanalizować mniejsze ich fragmenty (najlepiej kilka na jednym reprezentująca różne sytuacje obrazu). I porównać ich działanie dla różnych wariantów działania naszego algorytmu. W sumie na każdym wycinku do sprawdzenia na każdym są 4 warianty. **Pamiętać, żeby załączane obrazy były czytelne**, czyli nie załączać zrzutów ekranów tylko zapisane ploty i starajcie się tak je projektować żeby w PDFie nie uległy one pomniejszaniu, bo kompresja dokłada dodatkowe artefakty. Wycinki najlepiej, żeby były jednego rozmiaru najlepiej 128x128 lub 256x256.

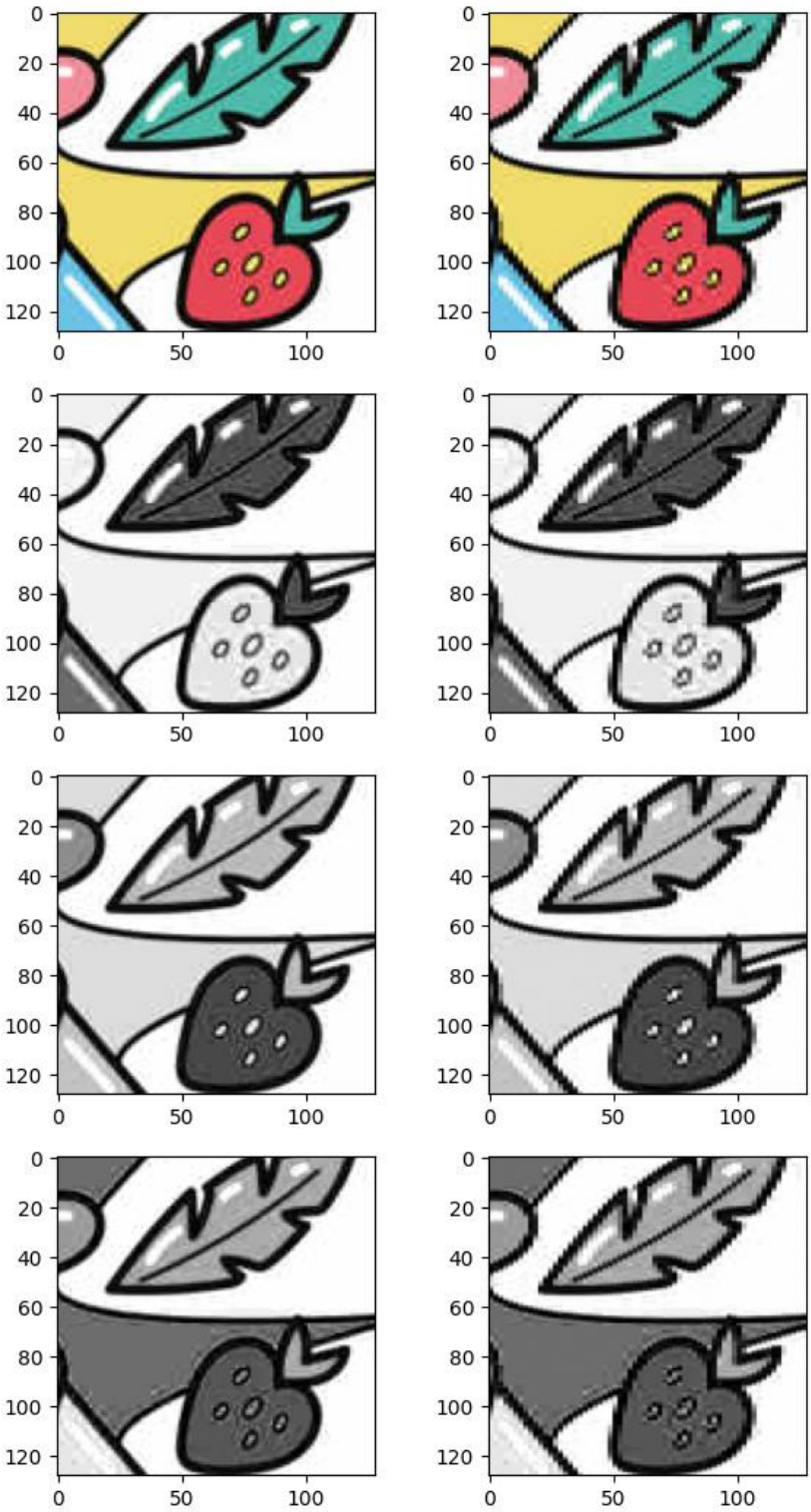
Wybrane zdjęcia:


WHAT's UP DOC?

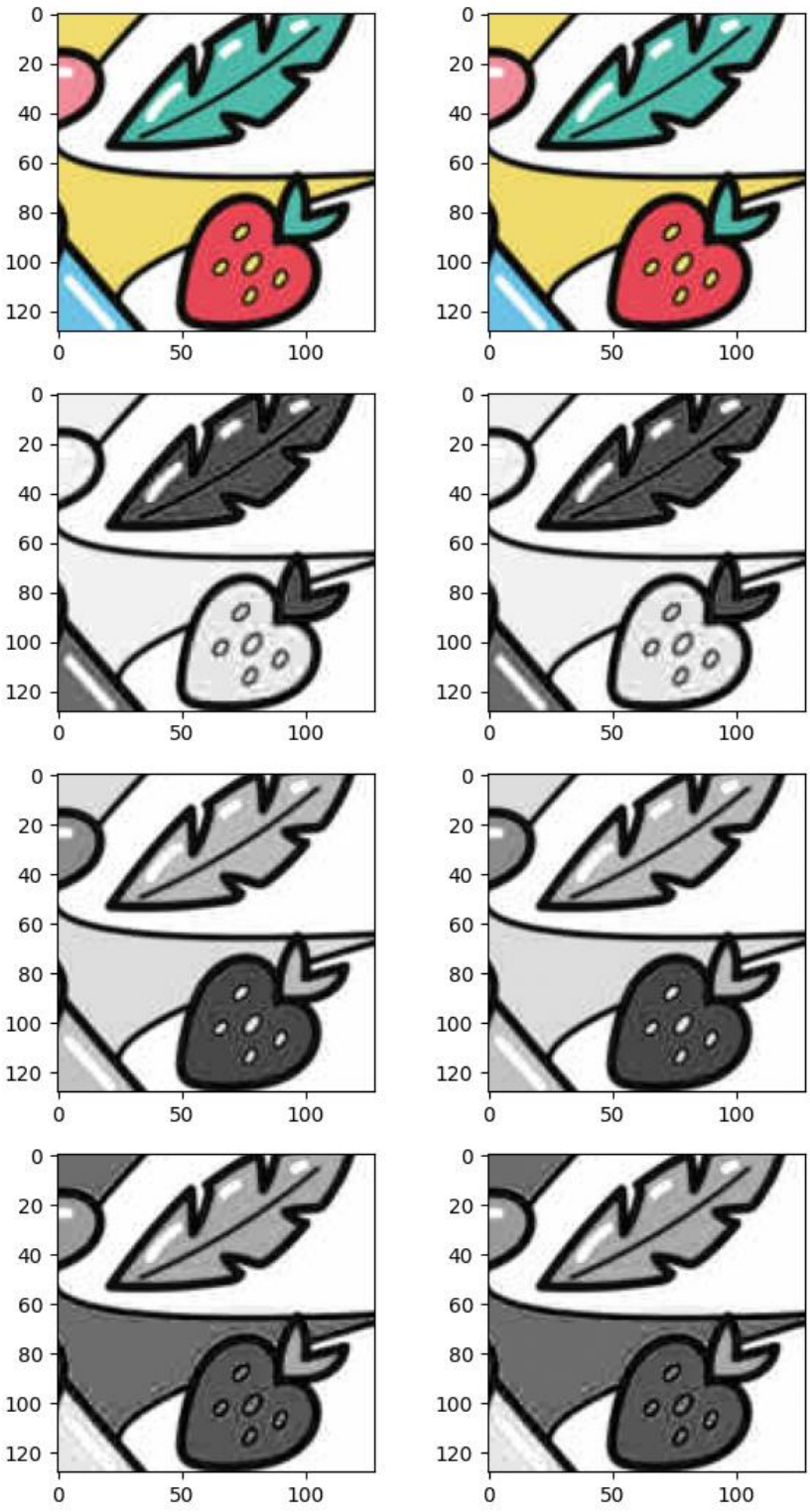
Krótkie wnioski:

Subsampling znacznie wpływa na wycinki, pojawia się pikselizacja. Przy 4:2:2 widac znaczącą pikselizacje, czego nie można powiedzieć o 4:4:4 Przy chrominacji obraz staje się nie wyraźny (lekko zaszumiony). Kompresja mocno przyczynia się do zniekształceń oryginalnych próbek. W każdym przedstawionym obrazku w tytule znajduje się rodzaj parametru subsamplingu oraz procent chrominancji (Kolejno na figure: oryginał, Y, Cr, Cb).

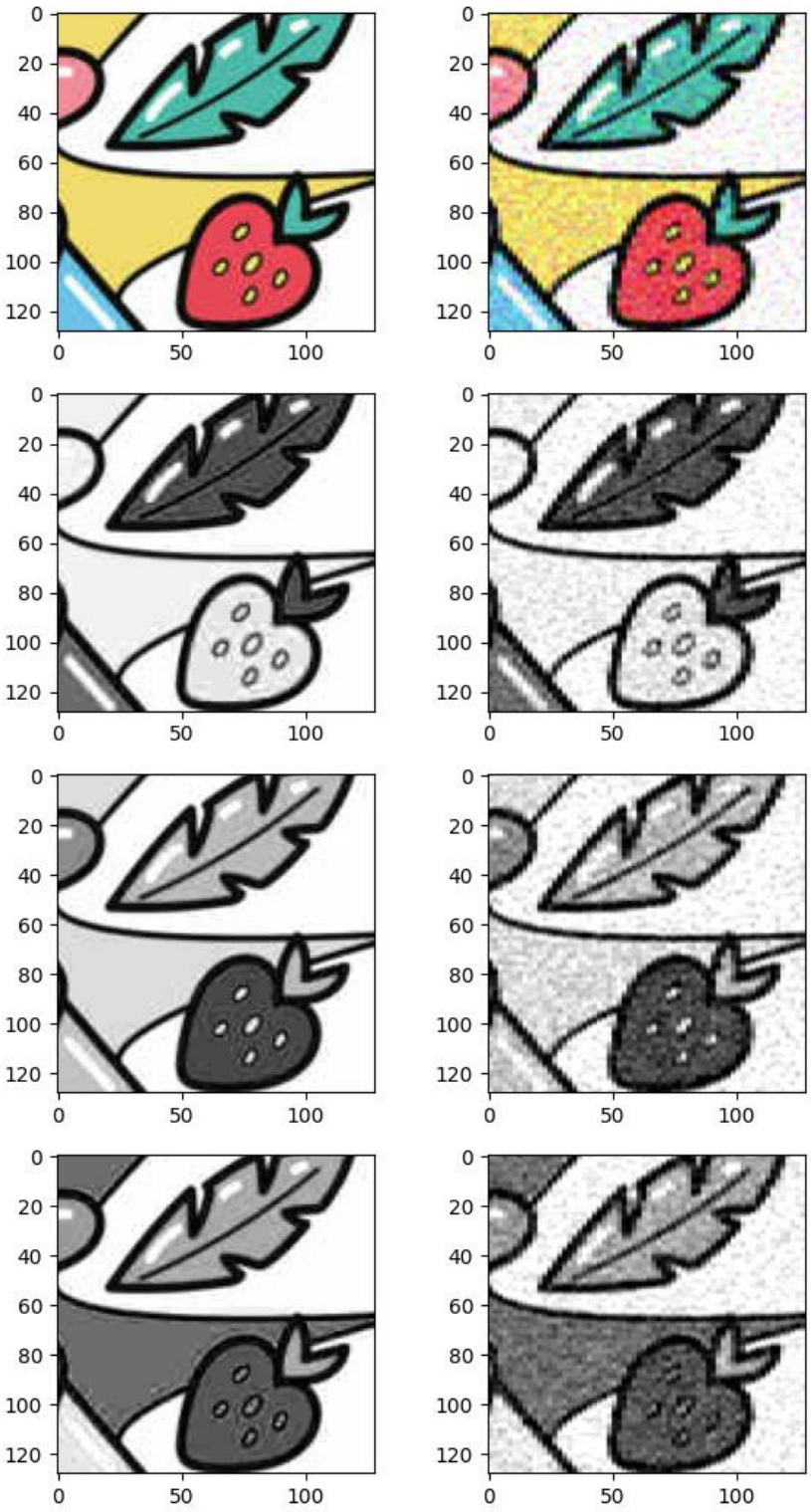
bug 4-2-2 100pro



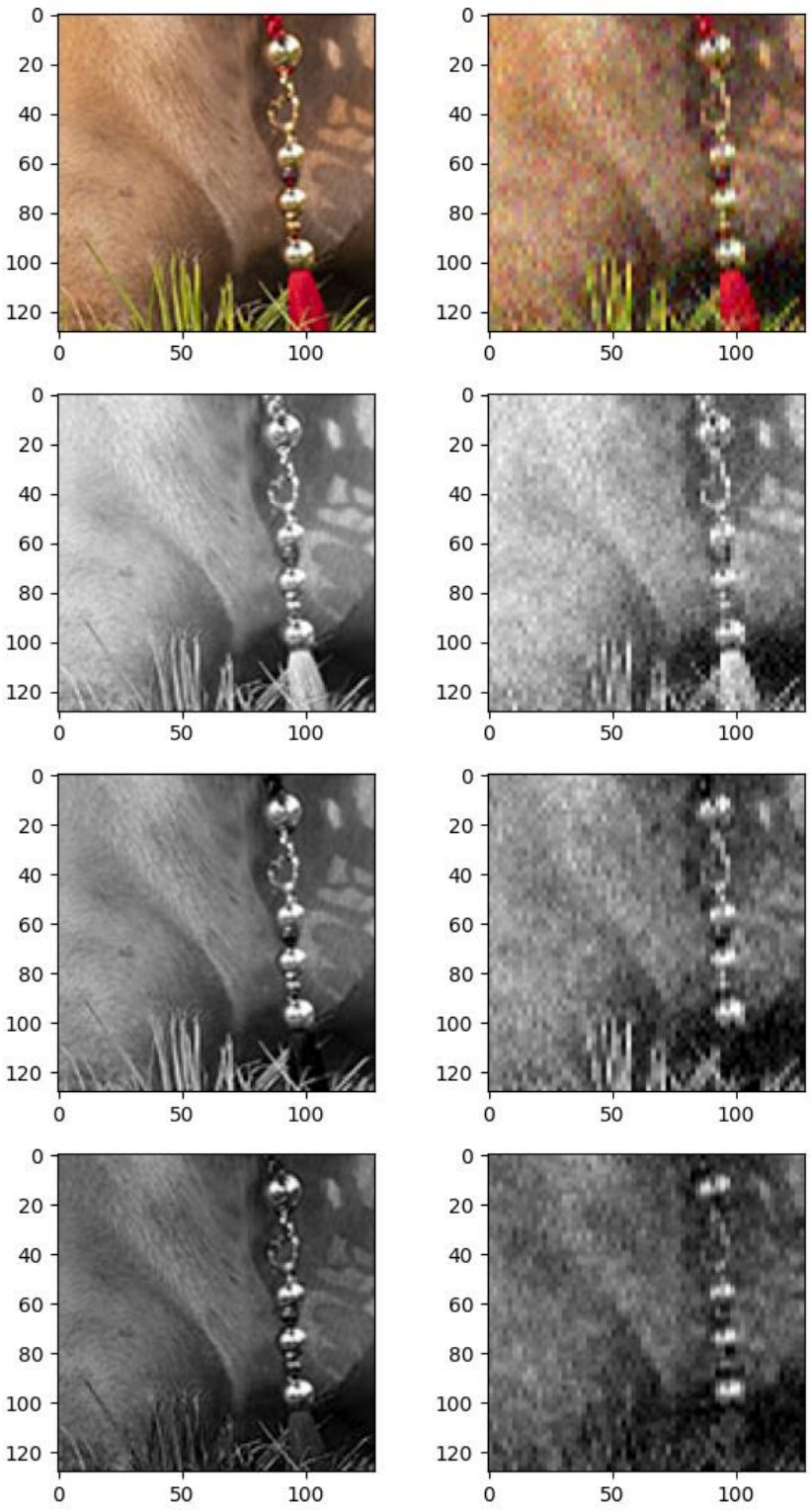
bug 4-4-4



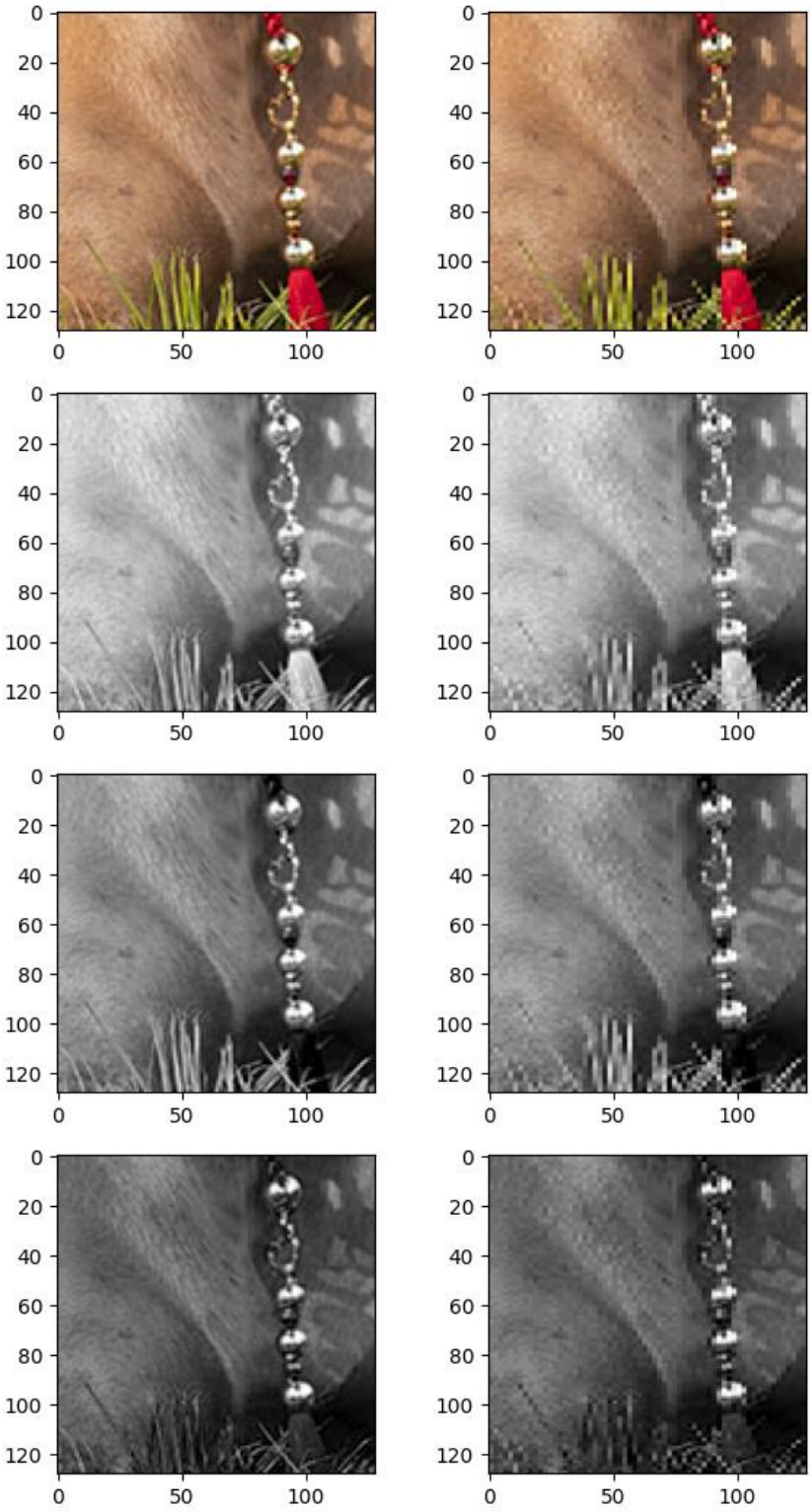
bug 4-2-2 50pro



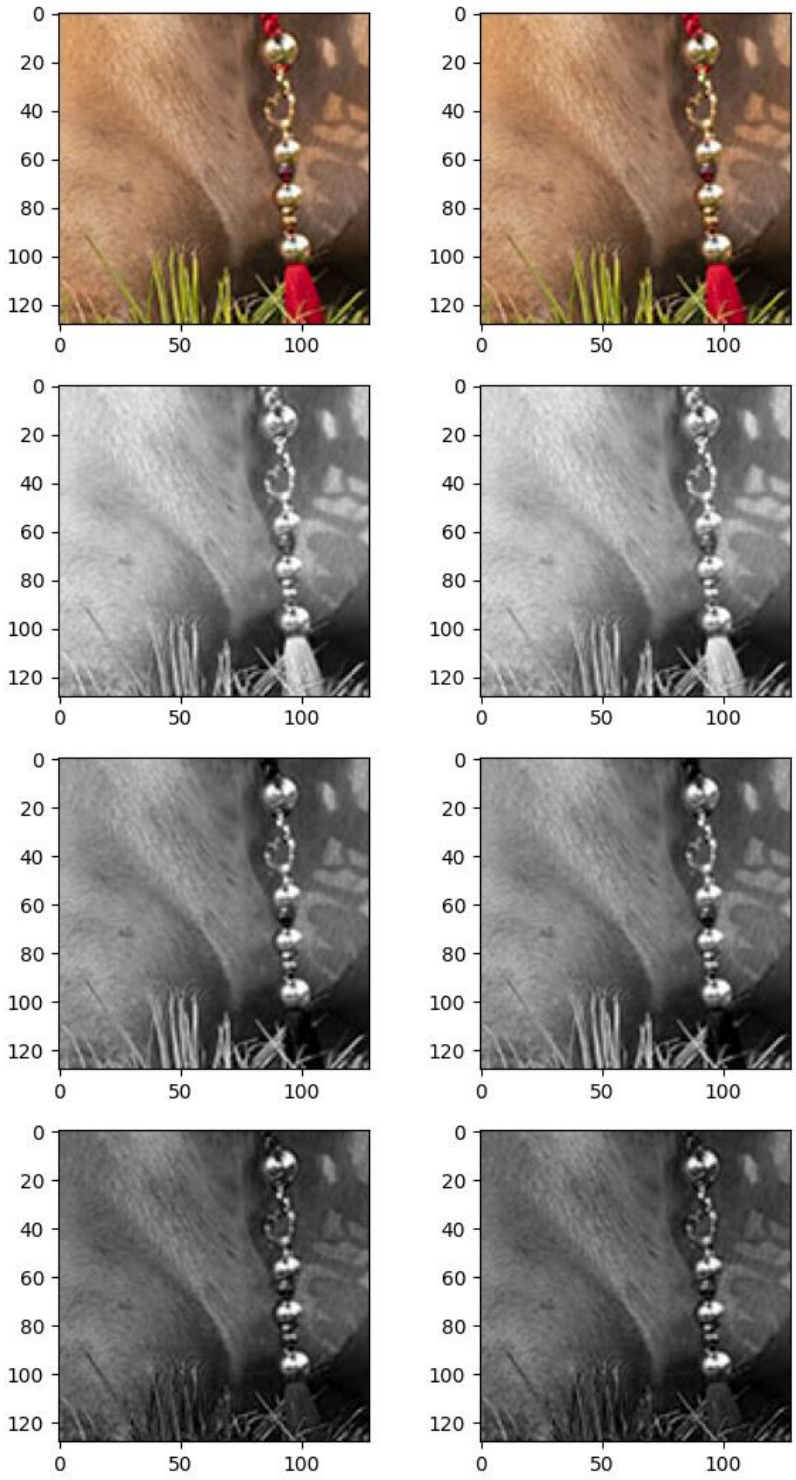
dog 4-2-2 50pro



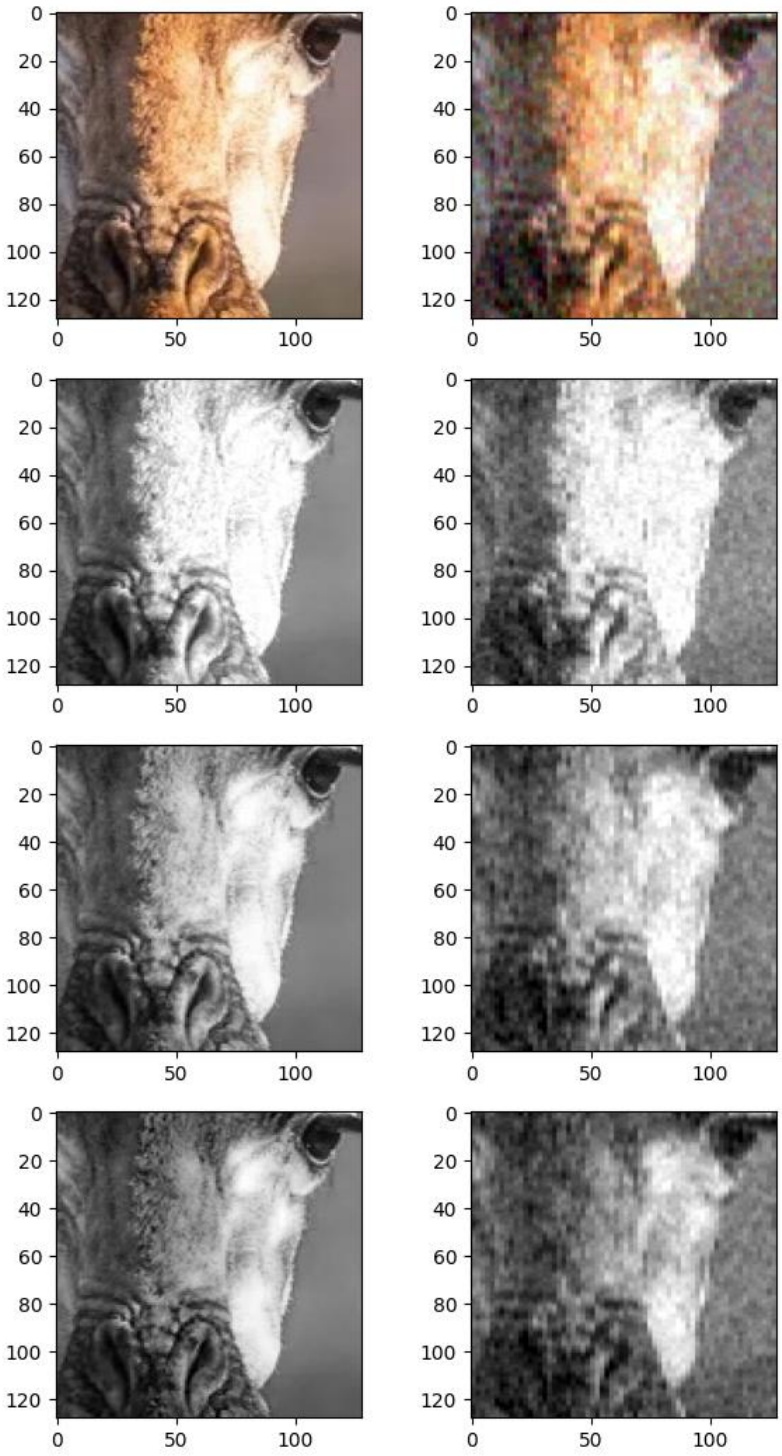
dog 4-2-2 100pro



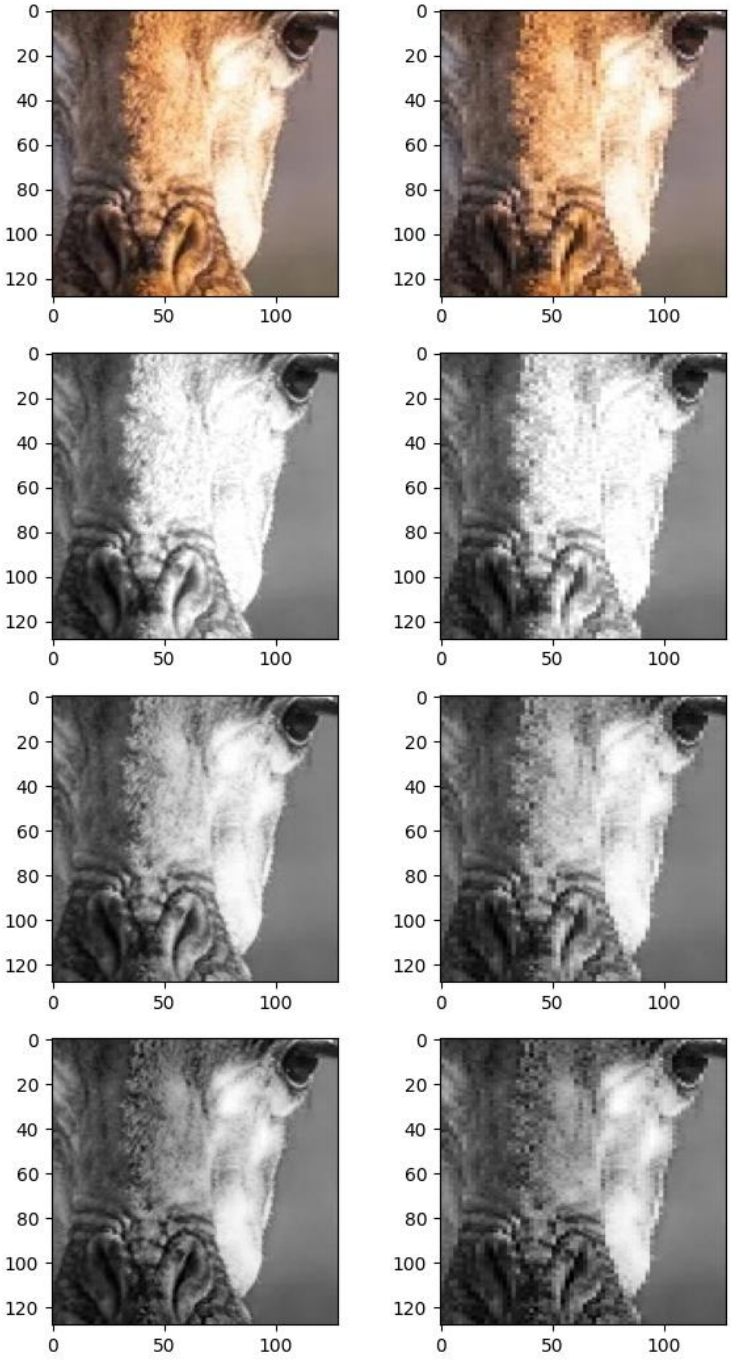
dog 4-4-4



zyr 4-2-2 50pro



zyr 4-2-2 100pro



zyr 4-4-4