

# **E-learningowy Projekt Zaliczeniowy**

## **Eksploracja Danych Internetowych**

Paweł Jeziorski, 234066      Karol Podlewski 234106

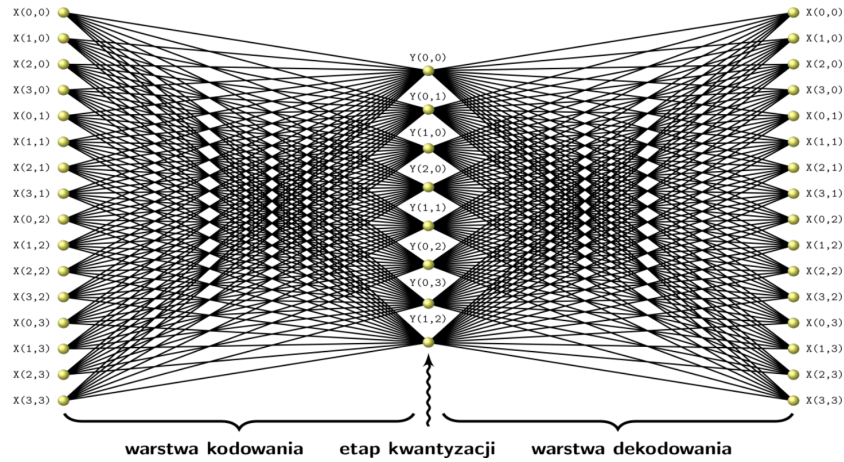
01.02.2021

# 1 Cel zadania

Celem zadania była implementacja sieci neuronowej typu autodekoder w celu kompresji obrazów.

## 2 Wstęp teoretyczny

Autodekoder to sieć neuronowa, która wykorzystywana jest przede wszystkim do ekstrakcji cech lub kompresji danych. Autodekoder to wielowarstwowy perceptron, składający się z neuronów liniowych, zawierający jedną warstwę ukrytą. Warstwy wejściowe oraz wyjściowe zawierają tyle samo neuronów, zaś warstwa ukryta zazwyczaj zawiera ich mniej - odpowiada ona za stratną kompresję danych.



Rysunek 1: Schemat sieci liniowej typu autodekoder [1]

W celu przetwarzania obrazów, należy pracować na ich pikselach. Obraz o wielkości 512x512 ładowany jest do pamięci właśnie w postaci tablicy dwuwymiarowej o takich właśnie rozmiarach. Żeby wprowadzić taką tablicę do sieci, należy ją najpierw spłaszczyć. Tworzenie wektora o 262 144 wejściach mija się z celem, dlatego operuje się na mniejszych fragmentach obrazu - np. w jednym momencie przetwarza się mniejszy fragment obrazu - kwadrat o wielkości 8x8 wymaga już tylko 64 neurony na wejściu oraz wyjściu.

Taki skompresowany obraz możliwy jest do oceny gołym okiem, ale jako dodatek przydatna może być także miara PSNR (ang. *Peak Signal to Noise Ratio*), którą wylicza się ze wzoru

$$PSNR = 10 \cdot \log_{10} \left( \frac{255^2}{\frac{1}{512^2} \sum_{i=0}^{511} \sum_{j=0}^{511} (x_{ij} - \hat{x}_{ij})^2} \right) \quad (1)$$

gdzie mianownik ułamka w logarytmie wynika bezpośrednio z największej możliwej wartości do uzyskania, a iteracje do 512 elementów są powiązane z wielkością przetwarzanego obrazu [1].

## 2.1 Implementacja

Zaimplementowana sieć neuronowa charakteryzuje się następującymi, zmiennymi parametrami:

- **Liczba neuronów w warstwie ukrytej** - w ramach zadania zmieniana będzie liczba neuronów w warstwie ukrytej - wartości, jakie przetestowaliśmy to 2, 4, 6, 8, 10, 12, 20 oraz 30,
- **Liczba iteracji** - przetestowaliśmy algorytm dla 10000 iteracji.
- **Współczynnik uczenia** - w omawianym rozwiązaniu ustawiony na stałe na wartość 0,01,
- **Szerokość wzorca** - długość boku, który będzie poddawany uczeniu w sieci. W wypadku tego zadania jest to 8. Długość boku definiuje także liczbę neuronów w warstwie wejściowej oraz wyjściowej - jest to 64, czyli tyle samo, ile pikselów we fragmencie obrazu o wielkości 8x8,
- **Liczba wzorców treningowych** - w przypadku omawianego rozwiązania zawsze jest to 10000.

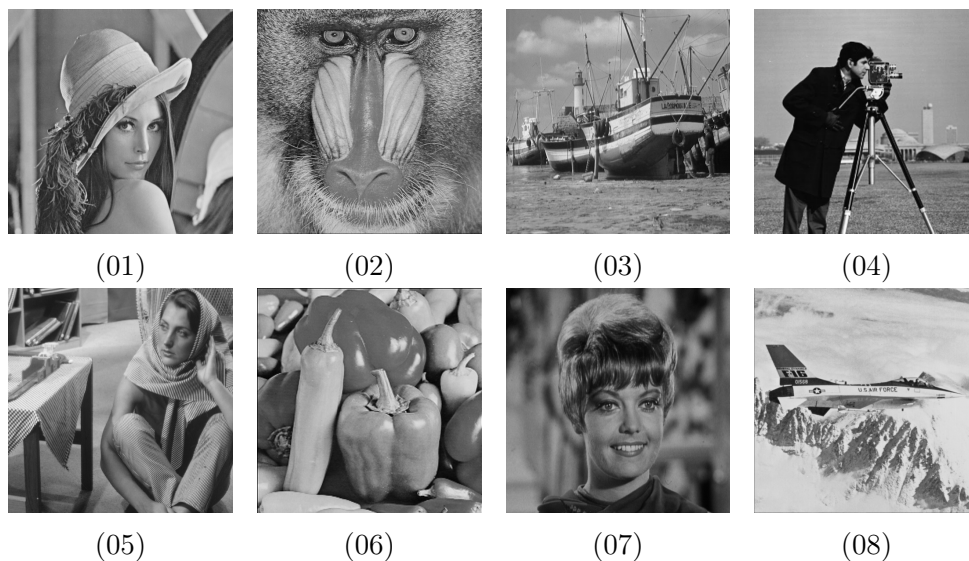
Wszystkie funkcje aktywacji są funkcjami liniowymi, a stopień kompresji obrazu obliczany jest ze wzoru:

$$CR = \frac{bp \cdot wysokosc\_obrazu \cdot szerokosc\_obrazu}{bw \cdot ni \cdot nh + bh \cdot no + bw \cdot nh \cdot no} \quad (2)$$

gdzie  $bp$  to liczba bitów na piksel (w wypadku przetwarzanych obrazów jest to 8),  $wysokosc$  i  $szerokosc$  obrazu jest równa 512,  $bw$  to bity potrzebne do zapamiętania wagi neuronu (we wzorze przyjęte zostało 8),  $bh$  odpowiada bitom potrzebnymi na zapamiętanie współczynnika na wyjściu warstwy ukrytej (przyjęte jako 12), a  $ni$ ,  $nh$  oraz  $no$  to liczba neuronów w warstwach wejściowej, ukrytej i wyjściowej - na potrzebę realizacji tego zadania liczba  $ni$  oraz  $no$  jest stała i wynosi 64.

### 3 Eksperymenty i wyniki

W ramach części doświadczalnej napisany program został sprawdzony na następujących 8 obrazach:



Rysunek 2: Obrazy wykorzystane w części eksperymentalnej.

#### 3.1 Opis eksperymentów

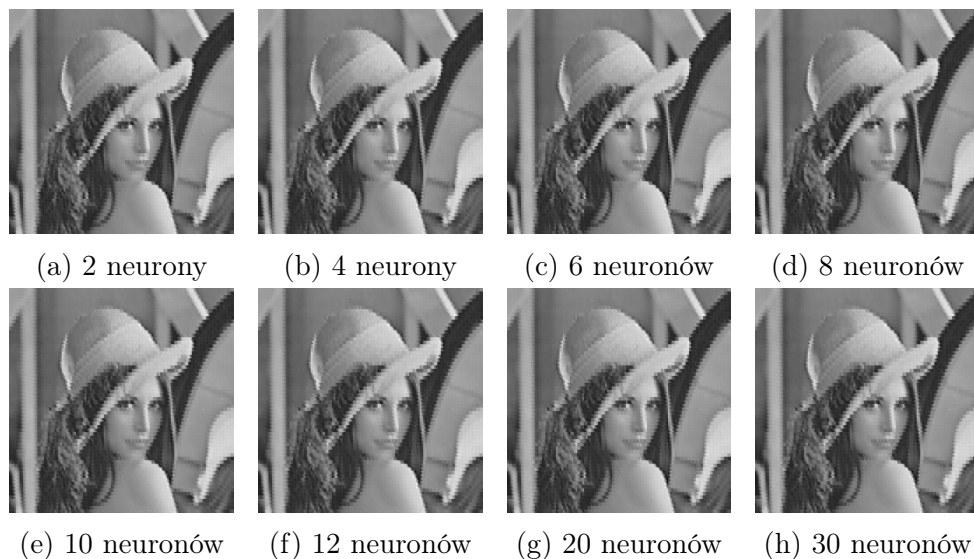
W ramach przeprowadzonych badań jedynym zmienianym czynnikiem były obrazy należące do grupy badawczej. Przeprowadzono 4 różne symulacje:

1. Zbiór treningowy składający się z obrazów 01 oraz 03,
2. Zbiór treningowy składający się z obrazów 02 oraz 04,
3. Zbiór treningowy składający się z obrazów 05 oraz 06,
4. Zbiór treningowy składający się z obrazów 06 oraz 08.

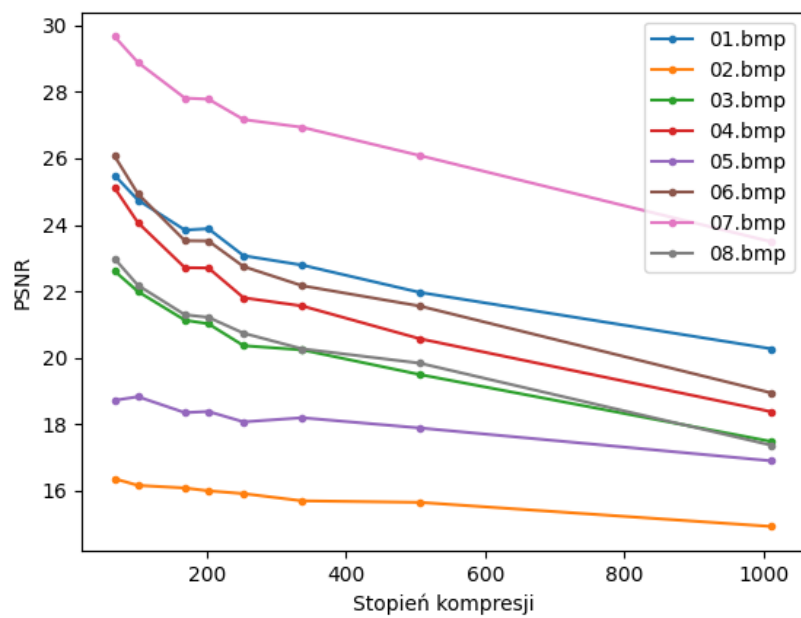
Wyniki uzyskanych eksperymentów przedstawiono w formie wykresów PSNR do stopnia kompresji.

## 3.2 Wyniki

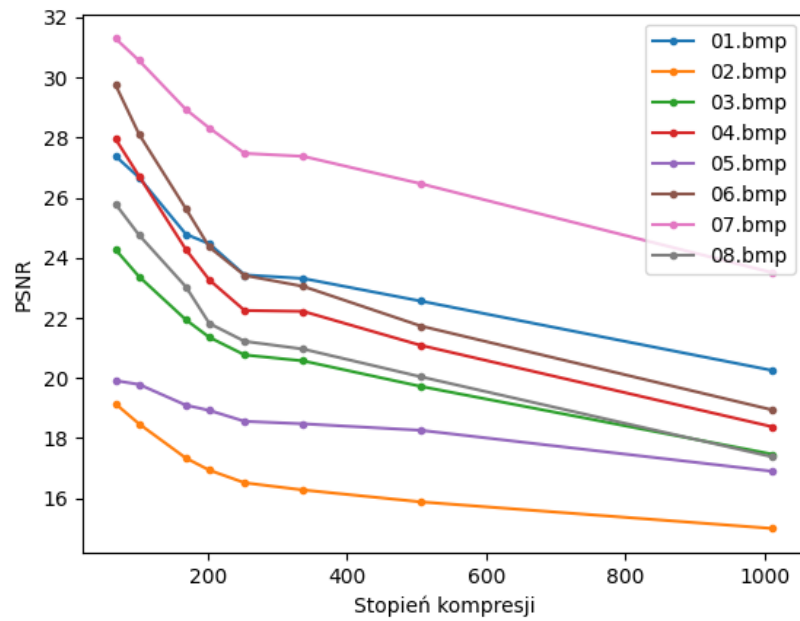
Przykładowa kompresja obrazu dla zestawu 1 znajduje się na Rysunku 3.



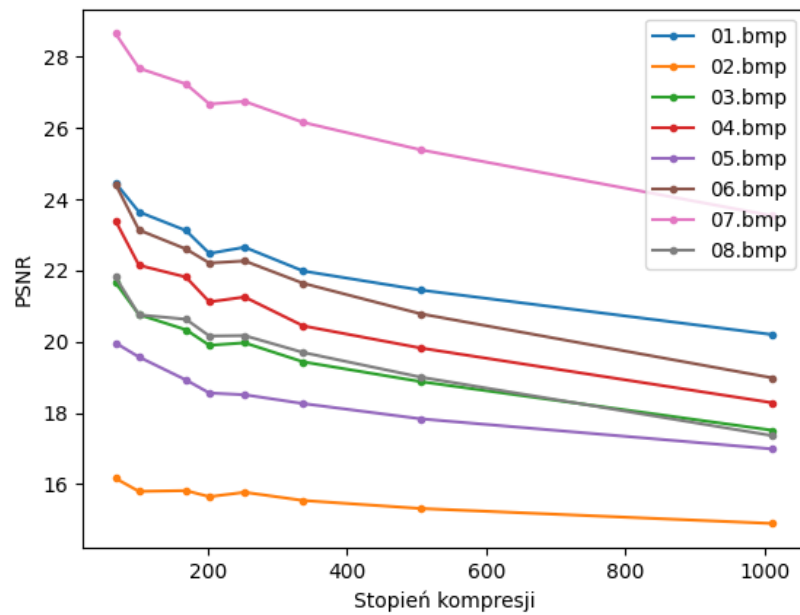
Rysunek 3: Kompresja obrazu A dla zestawu treningowego 1



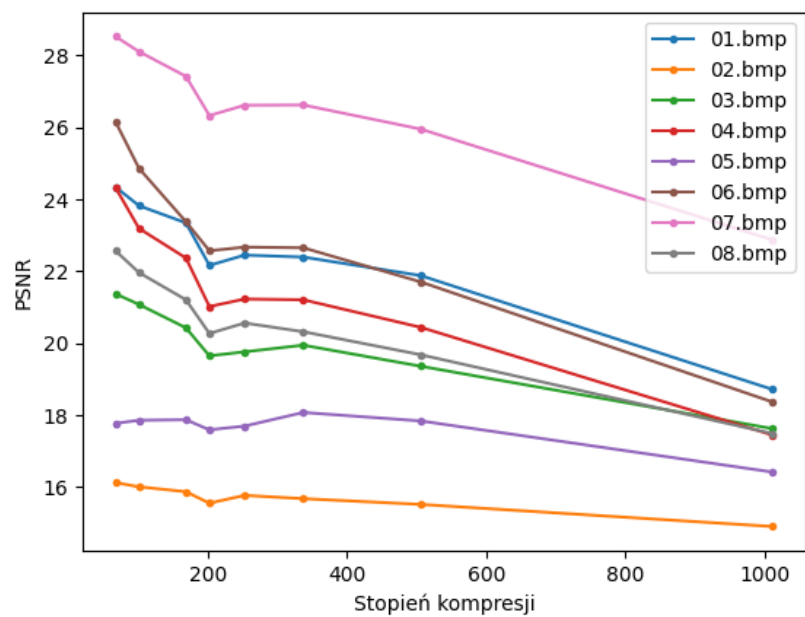
Rysunek 4: Wykres dla pierwszego eksperymentu



Rysunek 5: Wykres dla drugiego eksperymentu



Rysunek 6: Wykres dla trzeciego eksperymentu



Rysunek 7: Wykres dla czwartego eksperymentu

## 4 Dyskusja i wnioski

### 4.1 Dyskusja

Niezależnie od użytych obrazów do wytrenowania sieci, wszystkie z nich zachowywały się podobnie. Obrazy 02 i 05 zawsze cechowały się niską wartością PSNR (która w przypadku obrazu 05 przekroczyła 20 tylko dla najmniejszej kompresji przy treningu na niej samej). Najlepiej zachowywał się obraz 07, który potrafił przekroczyć wartość PSNR. Obrazy nigdy nie przekroczyły bariery 40 - zgadza się to z odczuciem wizualnym, gdyż zawsze można je odróżnić od oryginałów.

### 4.2 Wnioski

- Wybór obrazów do nauki nie wpływa znacznie na jakość kompresji - nauka na danym obrazie wciąż rzadko skutkuje dużo lepszą kompresją pod względem jakości wynikowego obrazka.
- Algorytm w swojej podstawowej wersji potrzebuje wielu iteracji oraz wielu wzorców, by dobrze kompresować obrazy - prosty krok nauki 0,01 może się okazać niewystarczająco dobry przy bardziej skomplikowanych zadaniach, przydatne może się okazać także skorzystanie z innych właściwości MLP.
- Sieci neuronowe okazują się bardzo przydatne przy kompresji obrazów.



## Bibliografia

- [1] Instrukcja do projektu e-learningowego.