

Koloryzacja zdjęć - raport

Norbert Baran

February 2022

1 O projekcie

Celem projektu było wytrenowanie modelu sztucznej inteligencji umiejętności przywracania rzeczywistych kolorów obrazu na podstawie jego czarno-białej wersji. Głównym zastosowaniem koloryzacji obrazów jest przywracanie oryginalnych kolorów starych fotografii.

Model nauczany był na podstawie datasetu *Image Colorization Dataset* udostępnianego przez portal www.kaggle.com:

<https://www.kaggle.com/aayush9753/image-colorization-dataset>

Repozytorium projektu dostępne jest pod adresem:

https://github.com/NorbertBaran/image_colorization

W skład repozytorium wchodzi następujące pliki:

- **README.md** - Krótki opis projektu.
- **image_colorization.ipynb** - Implementacja sieci neuronowej, na podstawie której wygenerowane zostały modele. Zawiera opis oraz niezbędne informacje o implementacji preprocessingu, procesu trenowania, testowania oraz wizualizację koloryzacji przykładowych obrazków, a także krótkie uzasadnienia zastosowanych rozwiązań.
- **colorizer_model_color_1.sav** - Wytrenowany dla 100 epoch model pierwszego z dwóch predykowanych kanałów koloru grafiki reprezentowanej w przestrzeni barw LAB opisanej w pliku *image_colorization.ipynb* oraz poniżej w sekcji *Algorytmika i przebieg implementacji projektu*.
- **colorizer_model_color_2.sav** - Wytrenowany dla 100 epoch model drugiego z dwóch predykowanych kanałów koloru grafiki reprezentowanej w przestrzeni barw LAB opisanej w pliku *image_colorization.ipynb* oraz poniżej w sekcji *Algorytmika*.
- **image_colorization_report** - Owy raport

2 Algorytmika i przebieg implementacji projektu

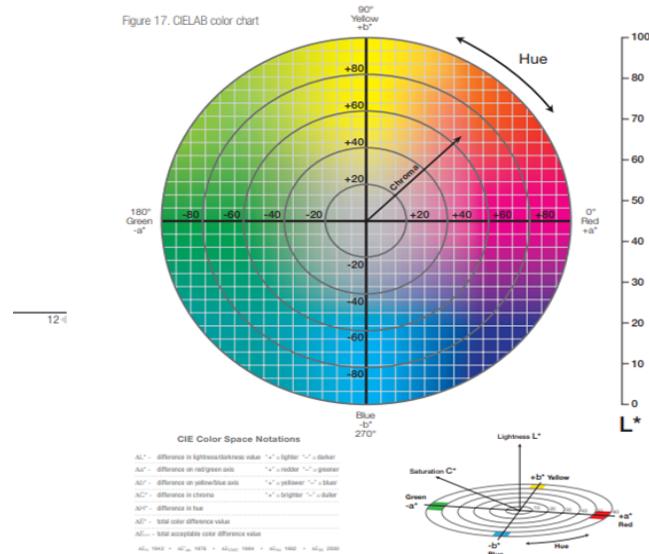
Pierwotnie zakładanym sposobem implementacji modelu nauczania maszynowego było podejście do problemu na dwa sposoby.

Pierwszy z nich przewidywał klasyczną metodę nauczania maszynowego jaką jest regresja liniowa oraz wykorzystanie dostępnej np. w pakiecie scikit-learn implementacji RandomSearchCV umożliwiającej sprawne przetestowanie dużej ilości kombinacji parametrów dla modelu regresji liniowej poprzez ich losowy dobór. Ze względu na poziom skomplikowania problemu, dla którego prymitywne dopasowanie odcieni szarości do skali kolorów okazało się mało skuteczne, podejście to nie było kontynuowane.

Druga koncepcja zakładała implementację głębokiej sieci neuronowej. W pierwotym założeniu miała to być gęsta sieć neuronowa. Podejście to również nie dało żadnych rezultatów.

Jako ostatecznie rozwijana, udostępniona w repozytorium projektu pod nazwą *image_colorization.ipynb*, wybrana została implementacja oparta na sieciach konwolucyjnych. W oparciu o nie zaimplementowano został autoenkoder, którego zadaniem jest dokonywanie translacji kanału szarego na kanały palety kolorów LAB.

Paleta kolorów lab:



Kolor reprezentowany przez paletę kolorów LAB posiada 3 kanały: pierwszy z nich przechowuje informację o jasności, natomiast pozostałe 2 oparte są na nieliniowo skompresowanych współrzędnych przestrzeni barw CIEXYZ. Paletę przestrzeni barw LAB przedstawia grafika powyżej.

Wykorzystując paletę barw LAB zamiast palety barw RGB do przewidzenia mamy 2 zamiast 3 kanałów koloru, gdyż dla palety barw LAB wartościami kanału jasności są szarości wejściowej grafiki. Podejście to znacznie poprawia wyniki względem palety barw RGB.

Ze względów praktycznych predykcje kanałów palety barw LAB nauczane są osobno. Są to 2 niezależne modele. Dzięki temu łatwiejsza i bezpieczniejsza jest kontrola nauczania każdego z kanałów, ponieważ podejście to umożliwia minimalizowanie funkcji kosztu każdego kanału z osobna. Rezultaty owego podziału zaprezentowane są w sekcji *Wyniki oraz wizualizacja*.

Skutkiem braku zasobów w postaci odpowiednio dużej pamięci ram oraz mocy obliczeniowej finalne modele nauczane były na datasetcie liczącym 25000 grafik oraz w ilości 100 epoch. Wygenerowane modele zamieszczone zostały w repozytorium, natomiast ich wykorzystanie można znaleźć w sekcjach *Testowanie* oraz *Wizualizacja* w pliku *image_colorization.ipynb*

3 Wyniki oraz wizualizacja

Początkowe wyniki na zbiorze treningowym:

Training color_1 first epochs:

Epoch 1/2

36/36 [=====] - 72s 974ms/step
- loss: 131.5444 - val_loss: 121.0836

Epoch 2/2

36/36 [=====] - 34s 961ms/step
- loss: 129.9672 - val_loss: 121.2457

Training color_2 first epochs:

Epoch 1/2

36/36 [=====] - 37s 974ms/step
- loss: 346.9572 - val_loss: 357.6230

Epoch 2/2

36/36 [=====] - 35s 961ms/step
- loss: 344.2183 - val_loss: 355.0281

Końcowe wyniki na zbiorze testowym dla datasetu 2500 obrazów oraz 100 epoch:

Evaluate color 1 on test data

24/24 [=====] - 13s 210ms/step
- loss: 112.4510

Evaluate color 2 on test data

24/24 [=====] - 4s 160ms/step -
loss: 284.3353

Wynikowe grafiki:

