

Zmiana stylu (artysty) obrazu z zachowaniem treści i semantyki

Sprawozdanie z projektu indywidualnego

Daria Danieluk

21.06.2023

1 Zapoznanie teorytyczne

W ramach projektu została wykorzystana literatura:

- Deep Learning. Praca z językiem Python i biblioteka Keras - Francois Chollet
- Deep learning i modelowanie generatywne. Jak nauczyć komputer malowania, pisania, komponowania i grania - David Foster

2 Klasyfikacja zbiorów

W ramach projektu została wykonana klasyfikacja zbiorów:

- MNIST
- CIFAR-10
- CIFAR-100

2.1 Środowisko Google Collab

Klasyfikacja była wykonywana w środowisku Google Colab. Modele sieci były budowane sekwencyjnie. Przy pierwszym podejściu udało się uzyskać wyniki:

- MNIST: dopasowanie: 0,981; funkcja straty: 0,0622
- CIFAR-10: dopasowanie: 0,7949; funkcja straty: 0,6185

Następnie zostały zmienione modele i zostały uzyskane wyniki:

- MNIST: dopasowanie: 0,9918 funkcja straty: 0,0287
- CIFAR-10: dopasowanie: 0,7959 funkcja straty: 0,6316
- CIFAR-100: dopasowanie: 0,3526 funkcja straty: 6,3025

W kolejnym kroku sprawdziłem modele przedstawione w repozytorium do książki "Deep learning i modelowanie generatywne". Sieć została zbudowana na podstawie modelu z funkcjonalnym interfejsem API. Otrzymane wyniki:

- CIFAR-10: dopasowanie: 0.7029 funkcja straty: 0.8993
- CIFAR-100: dopasowanie: 0.4159 funkcja straty: 2.3788

2.2 Środowisko uczelniane

Kod z pierwszej próby został uruchomiony również na środowisku uczelnianym. Otrzymane wyniki:

- MNIST: dopasowanie: 0,981; funkcja straty: 0,0622
- CIFAR-10: dopasowanie: 0,7949; funkcja straty: 0,6185

Wyniki sa takie same na obu środowiskach.

Następnie zostały wykonane dalsze próby klasyfikacji zbioru MNIST. Otrzymane wyniki: dopasowanie: 0,9925; funkcja straty: 0,0474. Zatem zaklasyfikowanych poprawnie zostało 9925 obrazów; a nie poprawnie 75 obrazów. W ramach sprawdzenia zostało przedstawione 9 poprawnie zaklasyfikowanych obrazów oraz 9 niepoprawnie zaklasyfikowanych obrazów.

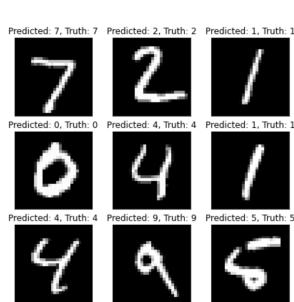


Figure 1: Poprawnie zaklasyfikowane

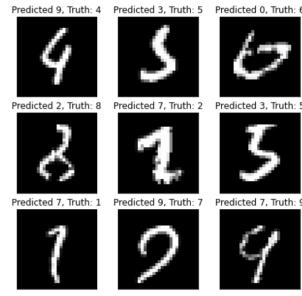
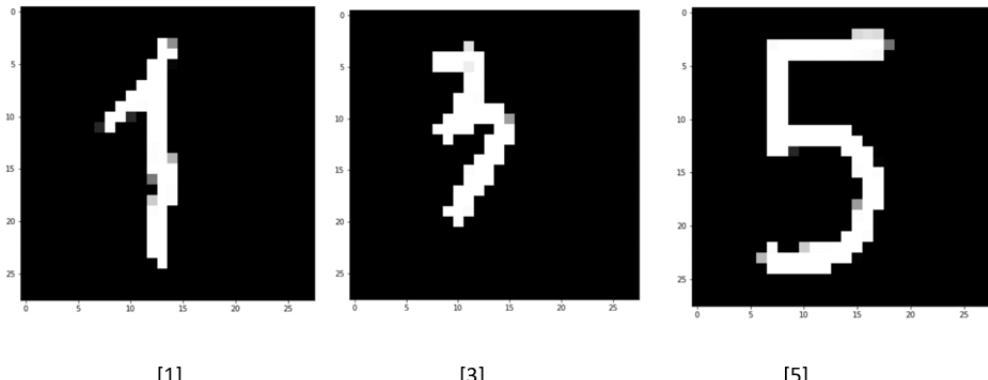


Figure 2: Niepoprawnie zaklasyfikowane

Następnie wytrenowany model został przetestowany na własnych obrazach.



Kolejnym krokiem była klasyfikacja zbioru CIFAR-10. Otrzymane wyniki: dopasowanie: 0,8250; funkcja straty: 0,5192.

Zbiór CIFAR-10 został przetestowany również z modelem z funkcjonalnym API w ramach którego udało się otrzymać wyniki: dopasowanie: 0.7228; funkcja straty: 0.8139.

Dla zbioru CIFAR-100 zostały uzyskane wyniki: dopasowanie: 0.3526; funkcja straty: 6.3025.

Modele przedstawione w repozytorium do książki "Deep learning i modelowanie generatywne" zostały także sprawdzone na środowisku uczelnianym. Otrzymane wyniki:

- CIFAR-10: dopasowanie: 0.7239 funkcja straty: 1.1711
- CIFAR-100: dopasowanie: 0.4159 funkcja straty: 2.3788

W ramach poprawy dopasowania na zbiorze CIFAR-100 zostały sprawdzone modele sieci: VGG oraz ResNet. Dla modelu wykorzystującego VGG uzyskałem dopasowanie 42%, dla modelu ResNet 67%.

3 Sieci CycleGAN

W ramach projektu zostało wykonane zapoznanie się z sieciami CycleGAN oraz próby zamiany obrazów jabłek na pomarańcze na zbiorze apple2orange.

Zamiana została wykonany w środowisku na Google Collab oraz na uczelnianym GPU.

Nie udało się zrealizować zamiany ze wzgledu na środowisko.

4 Neuronowy transfer stylu

Pierwsza próba polegała na uruchomieniu kodu ze strony keras.io gdzie wykorzystana była pętla która zmniejszała w każdej iteracji funkcje straty.



Figure 3: Obraz bazowy



Figure 4: Styl



Figure 5: Obraz po 100 iteracjach



Figure 6: Obraz po 4000 iteracjach

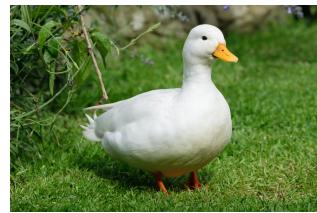


Figure 7: Obraz bazowy



Figure 8: Styl

Druga próba polegała na wykorzystaniu orginalnego algorytmu transferu stylu oraz przetrenowanego już modelu z TensorFlow Hub.



Figure 9: Wynik

5 Aplikacja pozwalajaca na zmiane stylu artysty

Projekt obejmował przygotowanie aplikacji z prostym interfejsem graficznym pozwalajacej na zamiane stylu obrazu na podstawie stylu artysty, poprzez połaczenie aplikacji z wyuczonym modelem sieci. Aplikacja została napisana w języku PyQt w środowisku PyCharm, wykorzystany został już gotowy przetrenowany model z repozytorium [Neural-Style-Transfer](#).

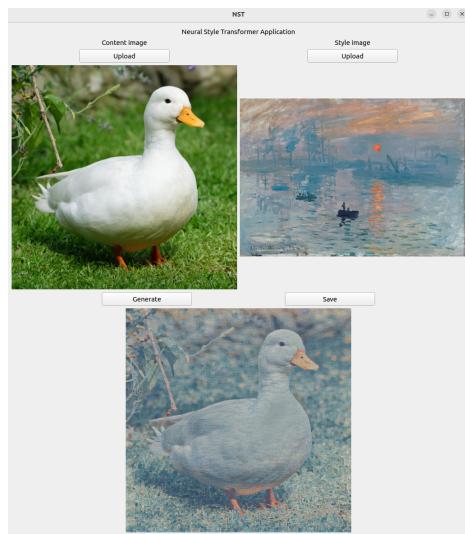


Figure 10: GUI

6 Sprawdzenie jak model zachowa sie na gładkich obszarach na zdjeciui

W ramach sprawdzenia zostało wykonana zamiana stylu na obrazie kota. Można zauważyć że futro zostało namalowane w bardzo podobnym stylu jak tło obrazu.



Figure 11: Obraz bazowy



Figure 12: Styl



Figure 13: Enter Caption