

Analiza Obrazów

Dokumentacja projektu

*Aplikacja przetwarzająca obrazy pod kątem zwiększenia
ich przyjazności dla osób ze ślepotą barwną*



Karol Mazurek

Ryszard Borzych

Przemysław Rewiś

Akademia Górniczo-Hutnicza w Krakowie

Wydział Fizyki i Informatyki Stosowanej

Opis Projektu

Celem projektu było stworzenie aplikacji do przetwarzania obrazów w taki sposób by były one bardziej przyjazne i czytelniejsze dla osób cierpiących na ślepotę barwną (w najbardziej pospolitych odmianach). Oprogramowanie powinno przyjmować na wejściu obraz i zwracać ten sam obraz po zmianie odpowiednich wartości kolorów.

Problem został rozwiązany w oparciu o sieć neuronową i zaimplementowany w programie MATLAB.

Narzędzia potrzebne do uruchomienia projektu

- MATLAB (R2018a) (wersja na taurusie)

Algorytmy stosowane w projekcie

W algorytmie wykorzystano trzy odrębne sieci neuronowe płytke o 20 neuronach w warstwie ukrytej (osobno dla każdego kanału).

W procesie uczenia do programu ładowane są 22 obrazy bazowe i te same obrazy z odpowiednio zmienionymi kolorami (kolory zmienione w Adobe Lightroom).

Obraz wejściowy i wyjściowy podzielono na fragmenty o rozmiarze 10 X 10 pikseli. W każdym z tych fragmentów jest obliczany bias o który należy skorygować wartości kolorów. Bias ten jest różnicą między uśrednioną wartością ze wszystkich pikseli we fragmencie:

$$bias = \frac{\Sigma out}{100} - \frac{\Sigma in}{100}$$

Jest to konieczny krok w celu zapewnienia odpowiedniej wydajności, a pojedyncze piksele nie mają znaczenia dla odbioru całego obrazu.

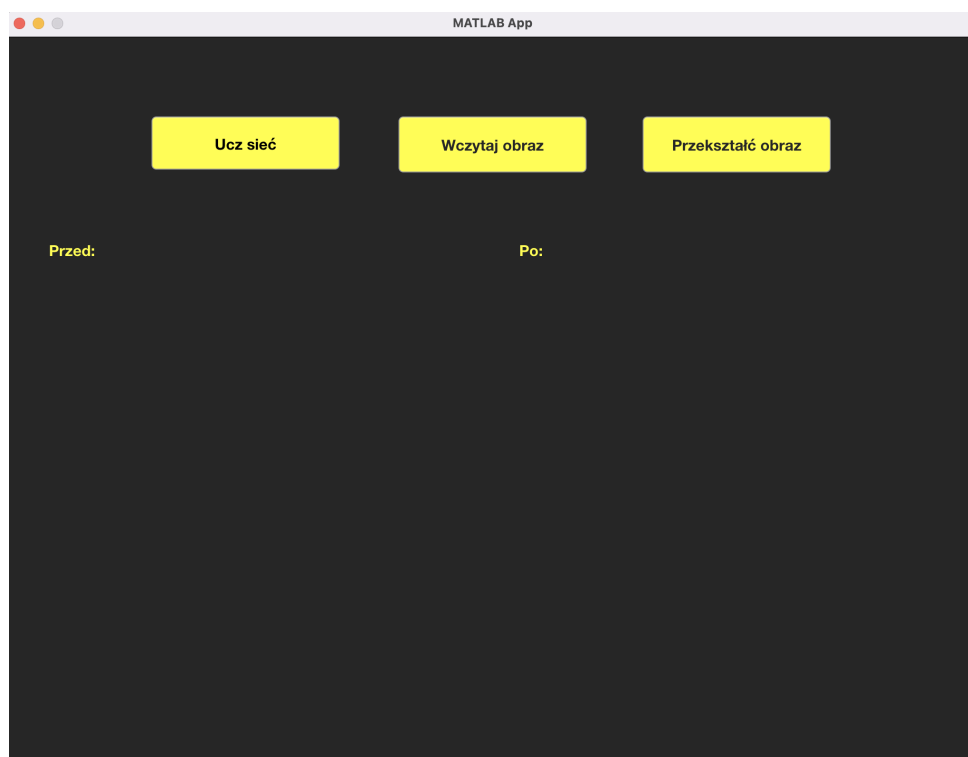
Następnie sieć stara się nauczyć wartości biasu dla zadanej wartości wejściowej. Do funkcji *train* podawany jest wektor zawierający wszystkie wartości biasu dla danego obrazu.

W celu poprawienia wydajności sieci po całym procesie uczenia do sieci są wpuszczane wszystkie wartości jakie obraz wejściowy może przyjmować. Następnie te wartości są stabilizowane i to one są używane do wyliczenia wartości obrazu wyjściowego (wartości są przybliżone do trzeciego miejsca po przecinku). Zabieg ten został zastosowany w celu przeniesienia czasu wykonywania na etap uczenia. Dzięki temu przetwarzanie obrazu 1600 X 1200 trwa zaledwie około 10 sekund. Odbywa się to kosztem jedynie niewielkiego spadku jakości.

Na wyjściu zwracany jest obraz ze wszystkimi kolorami powiększonymi o wartość biasu.

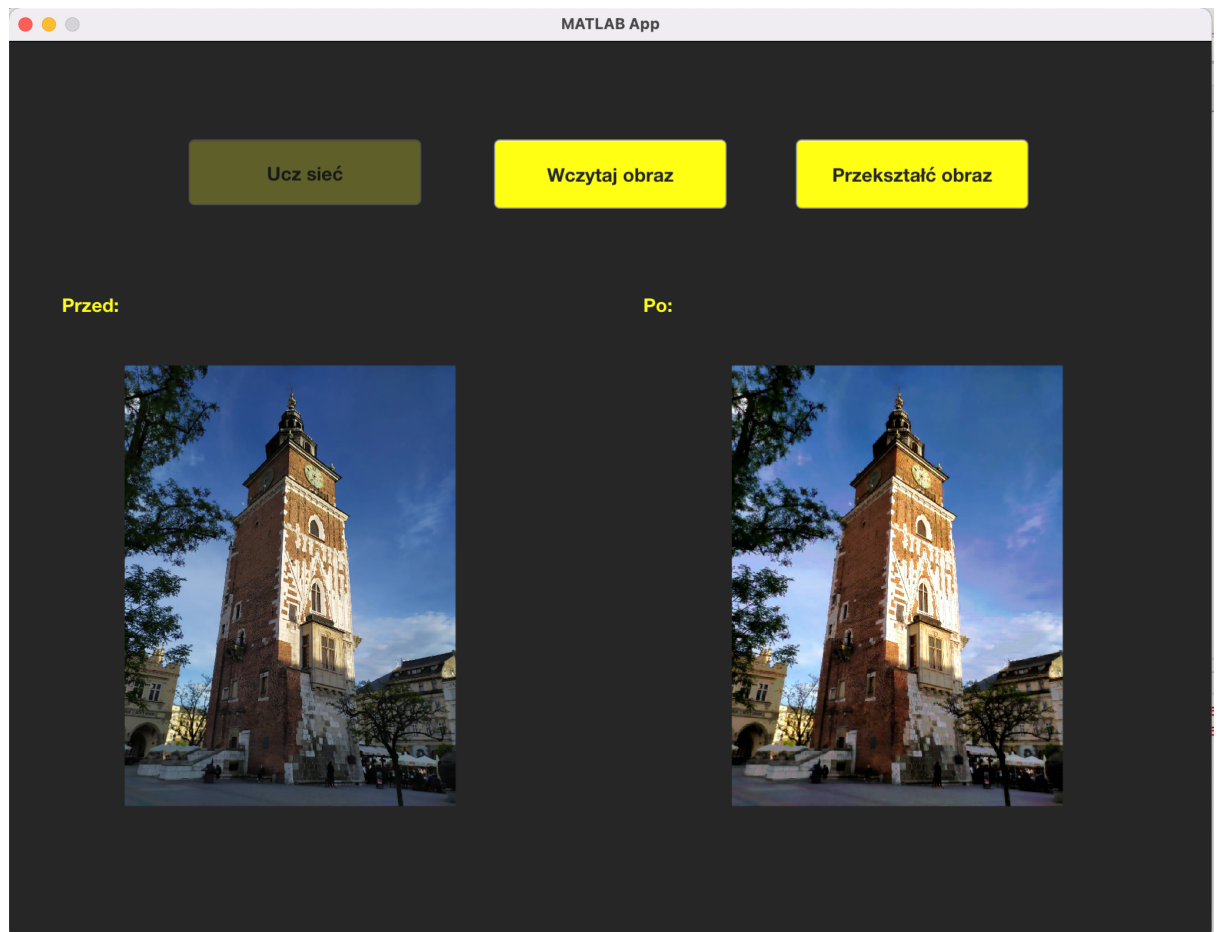
Sposób działania

Interfejs użytkownika został stworzony przy użyciu narzędzia App Designer, który automatycznie wygenerował większość kodu aplikacji.



Rys 1. Początkowe okno aplikacji

Przycisk “Ucz sieć” rozpoczyna proces trenowania sieci neuronowej. Przycisk “Wczytaj obraz” umożliwia wczytanie dowolnego obrazu w formacie .jpeg, .jpg, .png. Przycisk “Przekształć obraz” odpowiada za rozpoczęcie dostosowania.



Rys 2. Przykładowy wynik działania aplikacji.

Jak uruchomić

W celu uruchomienia aplikacji należy trzeba z poziomu matlaba kliknąć na plik projekt.mlapp dwa razy LPM. Otworzy się okno AppDesignera i po kliknięciu przycisku run u góry ekranu nastąpi uruchomienie aplikacji.

Najpierw należy kliknąć przycisk "Ucz sieć". Zostanie otwarte okno w którym trzeba nacisnąć OK, a następnie poczekać na zakończenie procesu uczenia.

Potem należy kliknąć "Wczytaj obraz" i wybrać odpowiedni obraz do podania przekształceniu (obraz musi mieć obydwie wielkości będące wielokrotnościami liczby 10).

Po wczytaniu należy kliknąć przycisk przekształć i po prawej powinien się pojawić obraz wynikowy.

Co NIE działa

W niektórych przypadkach zdarza się, że na obrazie wzdłuż prawej i dolnej krawędzi pojawia się ciemniejszy pas. Jest to uboczny skutek zwiększania wydajności sieci poprzez tablicowanie wartości.

Czasami może się nie wyświetlać poprawnie obrazek wejściowy, jest to problem z MATLABowym GUI. W takich sytuacjach można normalnie przeprowadzić proces uczenia i otrzymać obraz wyjściowy.

Wykonanie

- Karol Mazurek
 - przygotowanie obrazów wejściowych i wyjściowych do sieci neuronowej
 - doradztwo w kwestii poprawności działania sieci, ze względu na posiadanie ślepoty barw
 - dokumentacja
- Przemysław Rewiś
 - utworzenie GUI
 - dokumentacja
- Ryszard Borzych
 - zarządzanie siecią neuronową
 - dokumentacja