Rozpoznawanie tekstu

Projekt wykonany w ramach przedmiotu Analiza obrazów

Dawid Brach

Kacper Goraj

Mirosław Kołodziej

Michał Roguz

11.01.2022

1. Opis i założenia projektu

Głównym założeniem projektu było stworzenie programu rozpoznającego tekst z obrazów. Dodatkowo rozszerzyliśmy funkcjonalność o detekcję spacji w rozpoznanym tekście. Program został napisany za pomocą oprogramowania MATLAB w wersji R2020a, R2020b i R2021b.

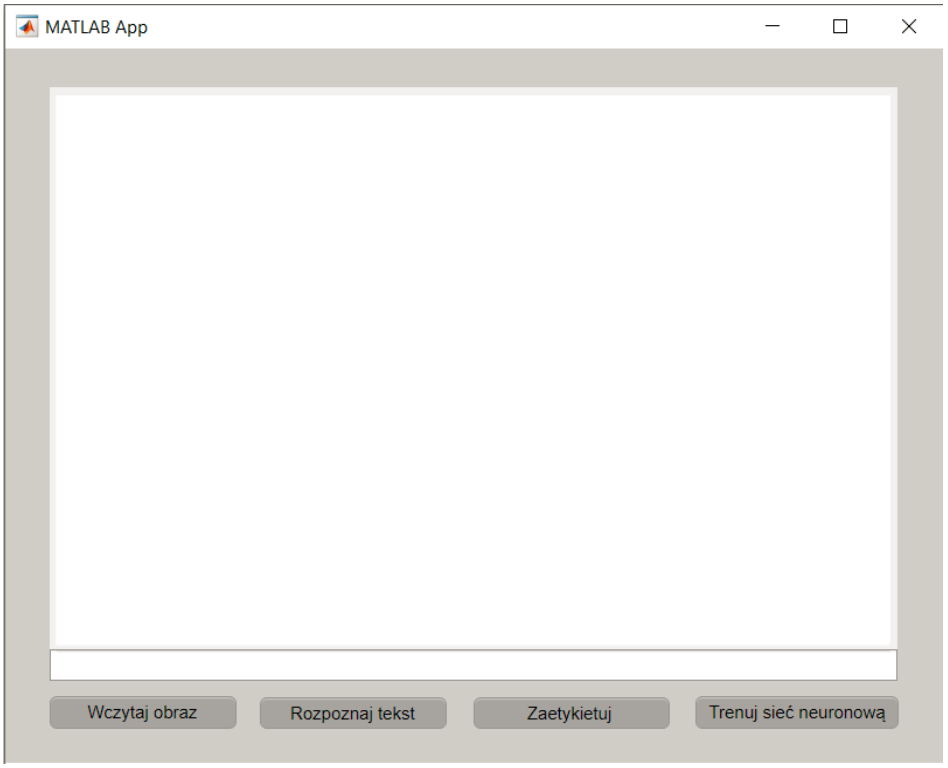
W projekcie wykorzystaliśmy kilka użytecznych toolboxów, które ułatwiły nam implementację. Zostały one wylistowane poniżej:

- Deep Learning Toolbox, który umożliwił nam wykorzystanie sieci neuronowych w projekcie,
- Image Processing Toolbox, umożliwiający obróbkę zdjęć,
- Computer Vision Toolbox, pozwalający na wyrysowanie ramek wokół rozpoznanych obiektów na obrazkach,
- text-to-speech, odpowiadający za wprowadzenie komentarzy głosowych.

2. Interfejs użytkownika

Interfejs użytkownika jest bardzo przyjazny dla klienta i zawiera następujące elementy:

- okno w którym pokazywany jest wczytany obraz,
- przycisk wczytujący obraz,
- przycisk rozpoznający tekst na obrazie i wypisujący go w oknie poniżej obrazka, dodatkowo zaznacza każdy obiekt na obrazie,
- przycisk do etykietowania obrazu, który wyodrębnia wszelkie obiekty, kolorując każdy na inny kolor
- przycisk służący do trenowania nowej sieci neuronowej



Rys. 1. Okno startowe programu

3. Działanie programu

3.1 Pierwsze uruchomienie

Program włączamy otwierając plik app1.mlapp. Wówczas powinno się pojawić okno MATLAB'a z GUI zaprezentowanym powyżej.

3.2 Przygotowanie danych przekazywanych do sieci neuronowej

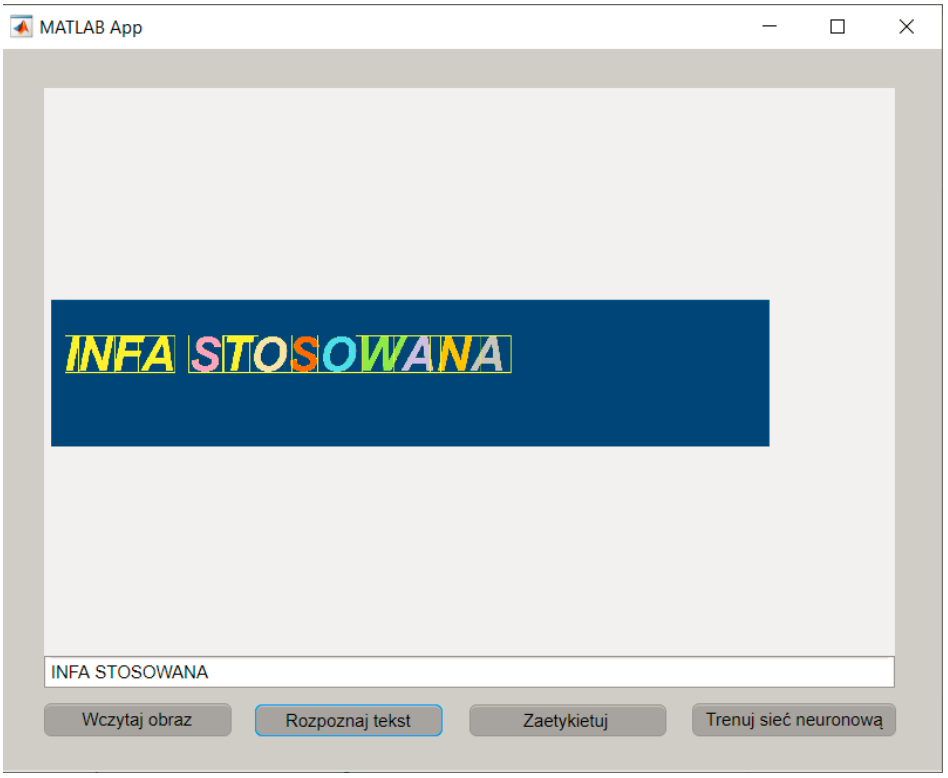
Na nasze dane składają się wielkie litery oraz cyfry czcionek Arial i Arial Narrow w 4 stylach (podstawowy, Bold, Italics, Bold&Italics). Znajdują się one w odpowiednich podkatalogach folderu Characters.

3.3 Algorytmy użyte w programie

W naszym programie zaimplementowaliśmy kilka algorytmów. Do łatwiejszych z nich należą binaryzacja oraz segmentacja i etykietowanie obrazu. Kolejnym z nich jest algorytm wyszukujący poszczególne litery z tekstu i obrysowujący je żółtą, prostokątną ramką. Uzyskuje on dane dotyczące położenia każdego znaku za pomocą funkcji regionprops. Następnie trzeba wspomnieć o wymyślonym przez nas algorytmie oddzielającym wyrazy spacją. Jego funkcjonalność polega na tym, że liczymy średnią odległość między poszczególnymi literami. Jeżeli różnica między sąsiednimi znakami jest większa od tej wartości, to wtedy program dodaje spację.

3.4 Obsługa programu

Wczytujemy obraz z tekstem naciskając przycisk “Wczytaj obraz”. Następnie klikamy przycisk “Rozpoznaj tekst”. Każdy znak zostaje zaznaczony a rozpoznany tekst zostaje wypisany na ekran. Dodatkowo lektor czyta wszystko, co zostało rozpoznane. Przycisk ten od razu binaryzuje obraz i filtruje operacją zamknięcia. Dzięki temu nasza aplikacja jest w stanie również rozpoznać tekst z kolorowych obrazków.

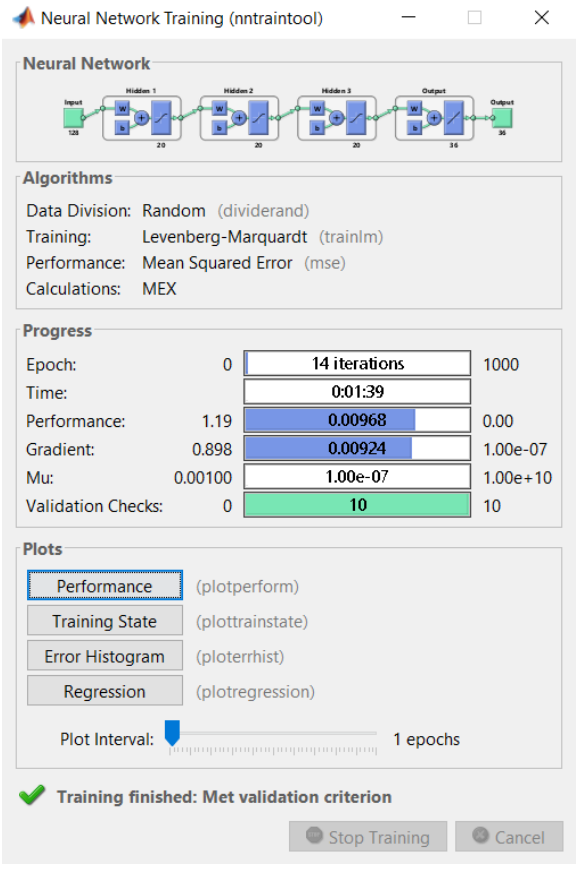


Rys. 2. Okno programu po wciśnięciu “Rozpoznaj tekst”

Funkcjonalność “Zaetykietuj” dokonuje segmentacji obrazu i koloruje wszystkie wyodrębnione obiekty.

Rys. 3. Okno programu po naciśnięciu przycisku "Zaetykietuj"

Ostatni przycisk ("Trenuj sieć neuronową") odpowiada za trenowanie nowej sieci neuronowej.



Rys. 4. Okno trenowania sieci neuronowej.

3.5 Sieć neuronowa

Sieć neuronowa została zaimplementowana przy użyciu funkcji feedforwardnet() z argumentem [20,20,20], do nauki zaś wykorzystaliśmy funkcję train() przekazując odpowiednio wynik z wcześniej wspomnianej funkcji, oraz wektor “trenowanie” oraz wektor “out”.

Wektor “trenowanie” składa się z danych do trenowania sieci neuronowej, zaś “out” to wektor odpowiednich wartości logicznych utworzonych funkcją repmat().

4. Podział pracy

Wszystkie prace wykonywaliśmy wspólnie podczas kilku spotkań online za pomocą aplikacji Discord. Tyczy się to zarówno pisania samego programu, jak przygotowania datasetu, trenowania sieci, testowania rozwiązania oraz tworzenia dokumentacji. W czasie spotkań jedna z osób pisała kod (udostępniając ekran), pozostałe zaś wymieniały się pomysłami i radami. Staraliśmy się, żeby za każdym razem pisał ktoś inny i rozkład pracy był równomierny.

5. Wdrożenie, raport i wnioski

Udało nam się zrealizować wszystkie założenia projektu, włącznie z dodatkowym zaproponowanym przez prowadzącego (detekcja spacji). Program powinien sobie poradzić nie tylko z obrazkami, które posiadają czarny tekst na białym tle, ale również z kolorowymi. Dodaliśmy również komentarze głosowe.

Zauważoną przez nas niedoskonałością jest problem z rozpoznawaniem niektórych liter po przekształceniu (za pomocą pogrubienia czy kursywy). Nie działają również litery pisane oraz te, posiadające zniekształcenia, więc aby program działał w pełni poprawnie zalecamy używanie wspomnianej wyżej czcionki. Kolejną niedoskonałością naszego programu jest marnowanie czasu przy powiadomieniach głosowych, np. po kliknięciu przycisku służącego do rozpoznawania tekstu najpierw czytany jest wygenerowany output, a dopiero następnie jest on wypisywany. Przy trenowaniu nowych sieci neuronowych zdarza się, że program rozpoznaje poprawnie mniej liter. Sieć, z której korzysta nasz program uzyskaliśmy dopiero po kilkudziesięciu próbach. Zalecane jest, aby jej nie zmieniać. Być może na różnych komputerach program działa inaczej, może to jednak zależeć od użytej wersji programu Matlab.

Jeśli chodzi zaś o rozszerzenie działalności programu lub usprawnienia, można poszerzyć bazę danych o np. małe litery, polskie znaki czy znaki interpunkcyjne. Kolejnym rozszerzeniem mogłoby być wprowadzenie większej ilości czcionek obsługiwanych przez program. Rozwiązaniem oszczędzającym czas użytkownika byłoby np. dodanie opcji wielowątkowości przy komunikatach głosowych. Jedną z najważniejszych zmian byłoby jednak usprawnienie trenowania sieci, gdyż dochodzi do tzw. stanu overfit, powodującym rozpoznawanie jedynie znaków użytych w bazie danych do trenowania.