

# Czytanie prostych równań matematycznych

Projekt zrealizowany w ramach przedmiotu Analiza Obrazów

Wykonali:

Mateusz Niepokój

Łukasz Wajda

Maciej Walczak

## 1. Założenia projektu

Założeniem projektu było stworzenie aplikacji umożliwiającej rozpoznanie prostych działań matematycznych z wczytywanego obrazka. Program został napisany przy użyciu oprogramowania MATLAB w wersji R2021b.

## 2. Opis funkcjonalności

### 2.1 Słowem wstępu

Program został napisany przy wykorzystaniu dwóch toolbox'ów Matlaba: *Deep Learning Toolbox* do wytrenowania sieci neuronalnej oraz *Image Processing Toolbox* do operacji na obrazach. Aby rozpocząć użytkowanie programu należy wybrać obraz zawierający wybrane przez nas wyrażenie. Program umożliwia jego odczytanie za pomocą wytrenowanej wcześniej sieci neuronowej, a następnie wskazanie wartości liczbowe lub narysowanie wykresu w przypadku podania wyrażenia ze zmiennymi 'x' oraz 'y'. Program pozwala również na wytrenowanie własnej sieci oraz jej przetestowanie. Kolejną funkcjonalnością jest zapisanie rozpoznanego wyrażenia do pliku .txt.

### 2.2 Działanie programu

#### Przygotowanie danych treningowych

Jako dataset, wygenerowane zostały wszystkie cyfry i litery 'x', 'y' dla kilkudziesięciu różnych fontów. Został użyty skrypt **generuj\_obrazy.m**. Skany dla każdego znaku przechowujemy osobno w podfolderach. Zdjęcia są przechowywane przy pomocy funkcji matlab **imageDatastore()**. Przygotowanie obrazów do uczenia sieci odbywa się w

funkcji **train\_nn.m**. Każdy zbiór obrazów jest przekazywany do funkcji **process\_img.m**, która dla każdego obrazu najpierw binaryzuje obraz a następnie liczy dla niego wektor współczynników kształtu. Na końcu zwraca zbiór tych wektorów jako macierz. Następnie takie macierze są zbierane w funkcji **train\_nn.m** do jednego wektora jako dane wejściowe sieci neuronowej.

Współczynniki liczone dla obiektów:

- Liczba Eulera
- Bottom Euler - mający na celu sprawdzenia liczbę eulera na przedziale 65% : 100% wysokości by wyłapać "6".
- Top Euler - mający na celu sprawdzenia liczby eulera na przedziale 0-35% wysokości.
- Współczynnik Malinowskiej
- Współczynnik kolistości S
- Współczynnik kolistości L
- Współczynnik Danielsson'a
- Współczynnik Kształtu
- Współczynnik Haralick'a
- Współczynnik Blair-Bliss'a
- Centroid1 - położenie środka ciężkości na osi x.
- Centroid2 - położenie środka ciężkości na osi y.
- ConvexArea - liczba pikseli obiektu.
- Extent - współczynnik powierzchni obiektu do tła.
- Eccentricity - określa stosunek między odległością punktu skupienia elipsy a długością jej osi dużej.
- Perimeter - określa obwód obiektu.
- Solidity

Kolejnym krokiem jest utworzenie poprawnych wyników dla tak określonych danych wejściowych. Wyjściem ma być macierz mająca kolumnę jedynek dla odpowiadającemu tej kolumnie znaku, cyfry lub litery. Najpierw tworzony jest wektor z jedną jedynką przekształcony przez funkcję **repmat()**. Tak otrzymane macierze są trzymane w wektorze jako spodziewane wyjście. Każda macierz odpowiada jednemu znakowi. Dzięki wektorowi trainin i trainout trenowana jest później sieć.

## Sieć neuronowa

Przy użyciu funkcji **feedforwardnet()** z toolbox'a MATLABA *Deep Learning Toolbox*, z argumentem '25' (oznaczającym rozmiar ukrytych warstw sieci) zaimplementowano sieć neuronową.

Do nauki sieci wykorzystano funkcję ***train()***, której argumentami były wynik funkcji ***feedforwardnet()*** oraz wektory ***trainin*** i ***trainout*** opisane w poprzednim paragrafie.

Dla odpowiedniego działania sieci ustawiono również inne parametry:

- `adaptFcn = 'adaptwb';`
- `divideFcn = 'dividerand';`
- `divideMode = 'sample';`
- `layers{1}.transferFcn = 'logsig';`
- `layers{2}.transferFcn = 'tansig';`
- `trainFcn = 'trainlm';`
- `performFcn = 'mse';`

Zestaw współczynników obliczonych przy użyciu sieci został zapisany do pliku "new\_neural\_network.mat", który wykorzystywany jest do późniejszego porównania ze współczynnikami obrazu załadowanego przez użytkownika na przycisk "Rozpoznawanie tekstu".

## Rozpoznawanie tekstu

Po załadowaniu pliku przez użytkownika, obraz poddawany jest takim samym operacjom (opisanym w paragrafie *Przygotowanie danych treningowych*) jak obrazy z datasetu wykorzystanego do nauki sieci.

W skrócie: obliczone współczynniki dla każdego z obiektów znalezionych na obrazie zostają porównane ze współczynnikami wyznaczonymi dla każdego znaku, dla której sieć została uprzednio nauczona. Odpowiedni znak zwracana jest przy największym możliwym dopasowaniu.

## 2.3 Dodatkowe funkcje

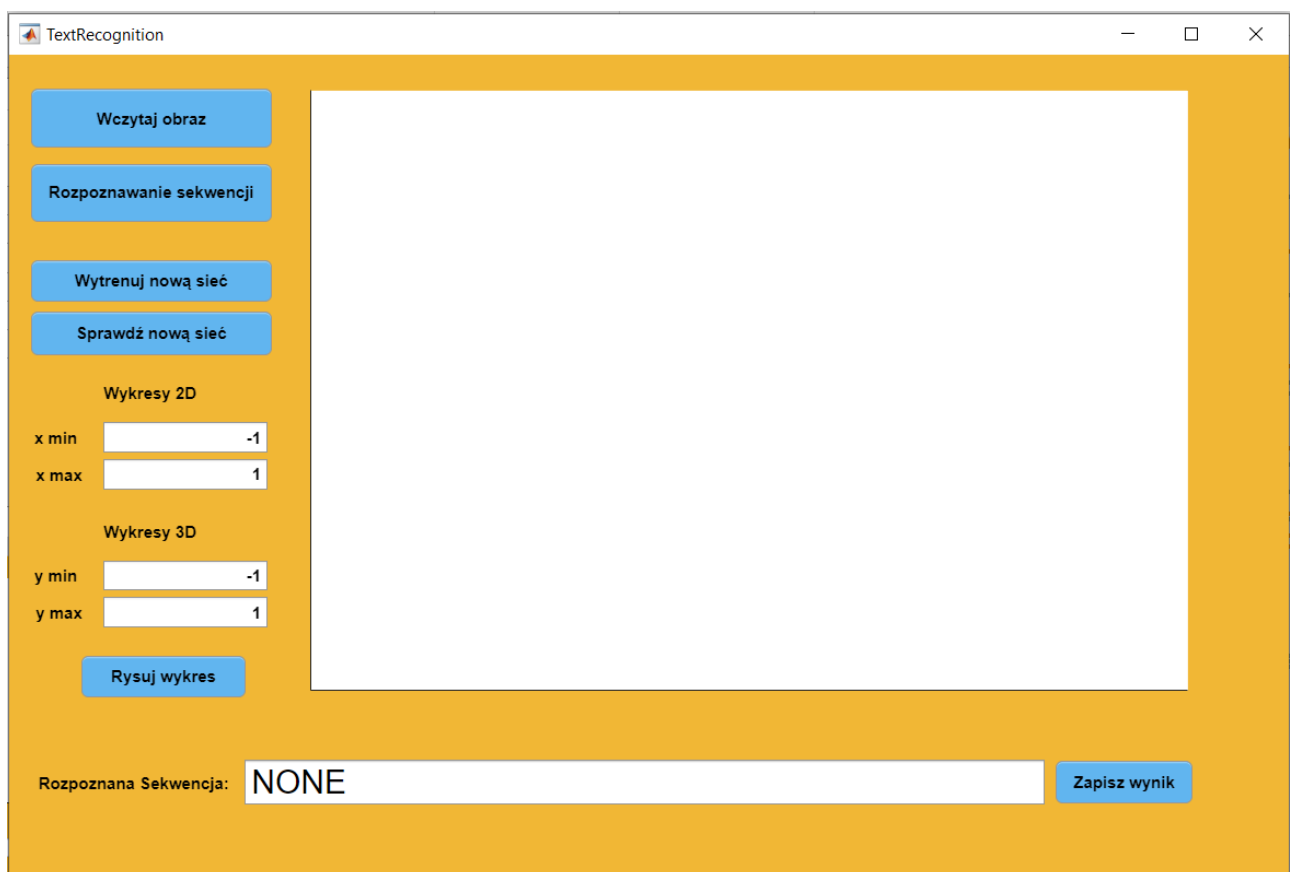
W ramach urozmaicenia aplikacji dodano możliwość ponowne wytrenowanie sieci neuronowej, rozpoznanie równania przy użyciu nowej sieci (za pomocą przycisku "Sprawdź nową sieć"), zapis wyniku rozpoznanego równania do pliku tekstowego, jak i

rysowania funkcji w 2 lub 3 wymiarach (w zależności od ilości zmiennych) dla wcześniej określonych zakresów [min;max] dla "x" i "y".

### 3. Interfejs użytkownika

Interfejs graficzny projektu został zrealizowany przy pomocy aplikacji MATLAB App Designer. W celu uruchomienia aplikacji należy otworzyć plik "EqRec.mlapp" z listy znajdującej się w głównym katalogu matlaba. Projekt uruchamiamy naciskając przycisk Run (zielona strzałka znajdująca się w środkowej części paska narzędzi)

W przypadku powodzenia powinien pojawić się poniższy ekran.

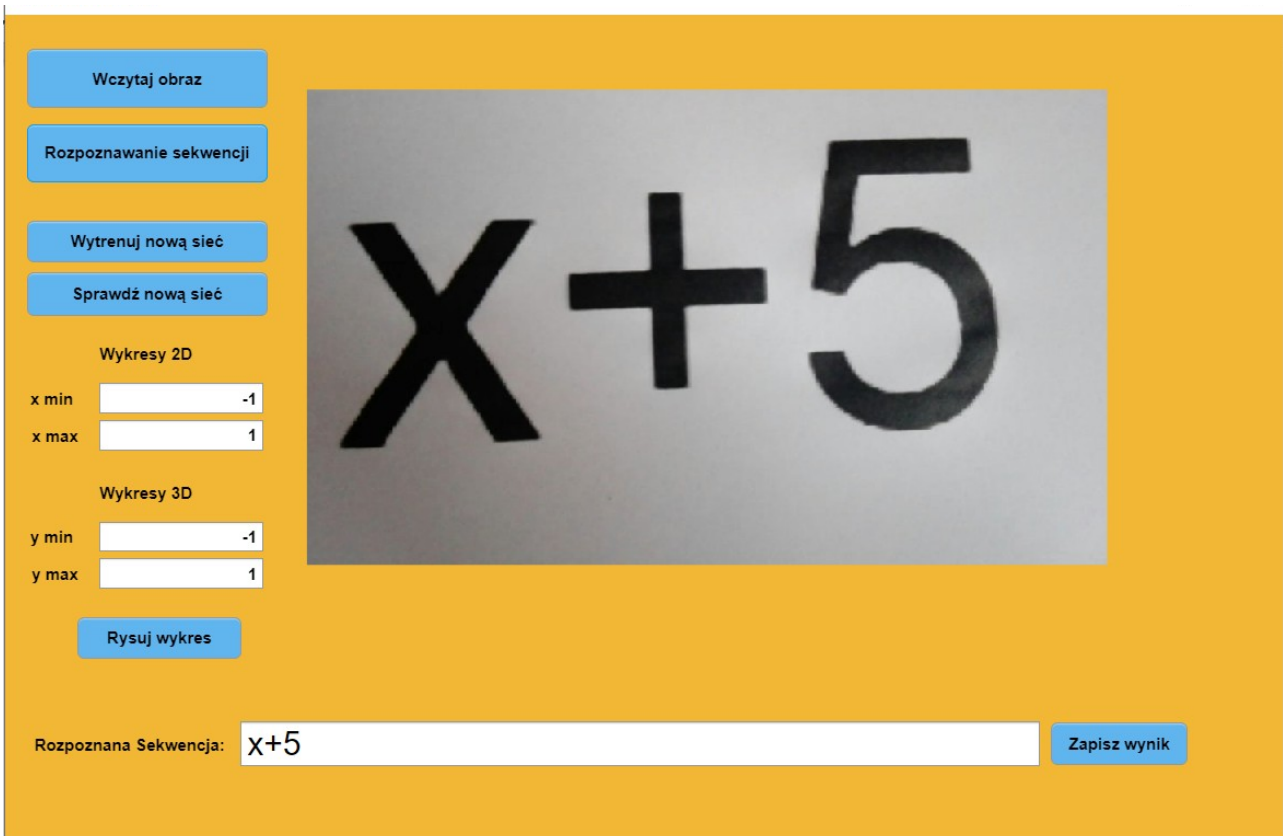


*Rysunek 1: Ekran startowy aplikacji*

W ramach GUI dostępne jest kilka przycisków. Do wczytania obrazu należy wykorzystać przycisk 'Wczytaj obraz'. Zaleca się, aby wczytywany obraz zawierał równania w kolorze czarnym napisany czcionką podobną do Ariala na białym tle. Do projektu zostały dołączone przykładowe obrazy o następujących nazwach:

- działanie1.jpg
- działanie2.jpg
- działanie3.jpg

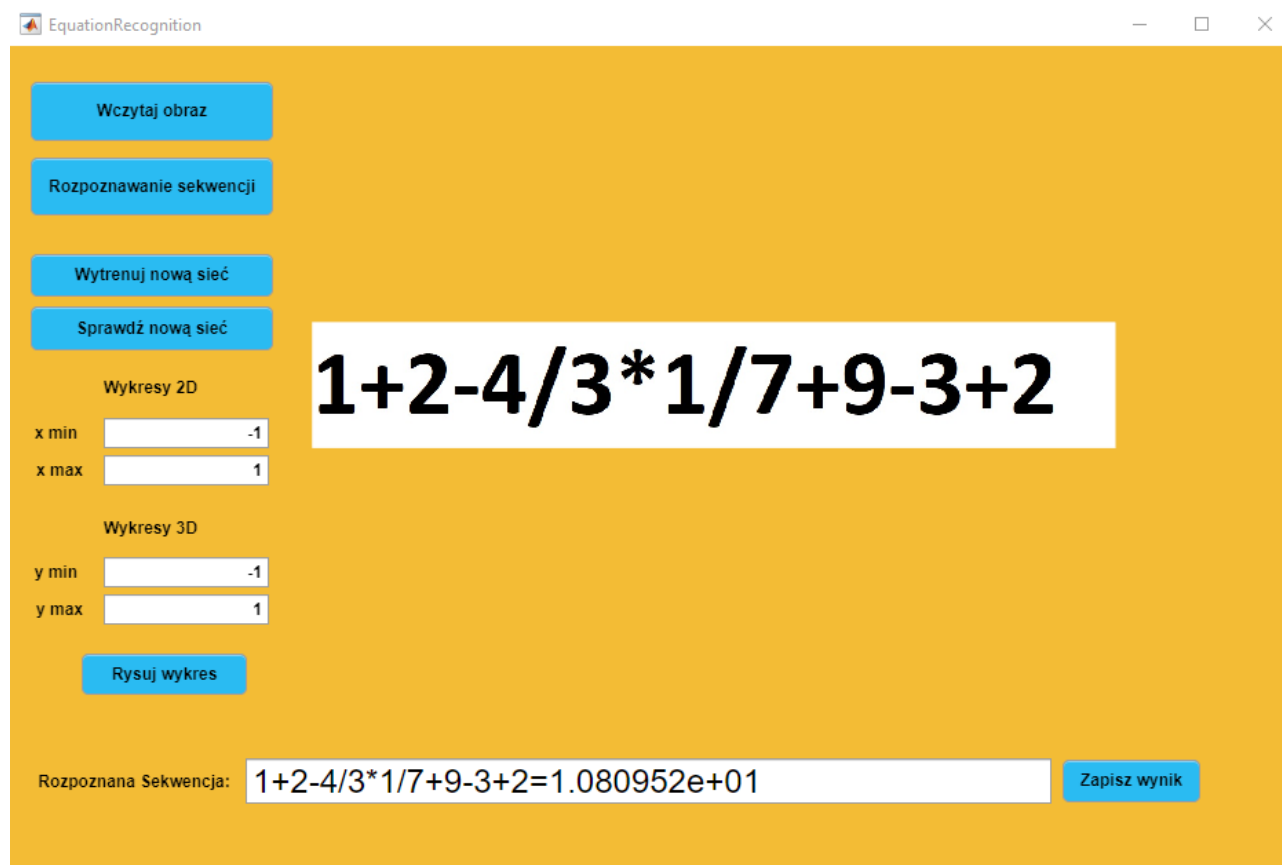
Po wybraniu i wczytaniu obrazu z pliku, powinien pojawić się on w lewej części ekranu, tak jak na poniższym obrazku.



*Rysunek 2: Ekran po wczytaniu obrazu z pliku*

Przycisk rozpoznawanie tekstu wykorzystuje wytrenowaną sieć neuronalną do rozpoznania znajdujących się na obrazie znaków. Do jego działania wymagane jest, aby był wczytany obraz. Program obsługuje cyfry [0-9], operacje matematyczne '/', '\*', '-', '+' oraz litery 'x' oraz 'y';

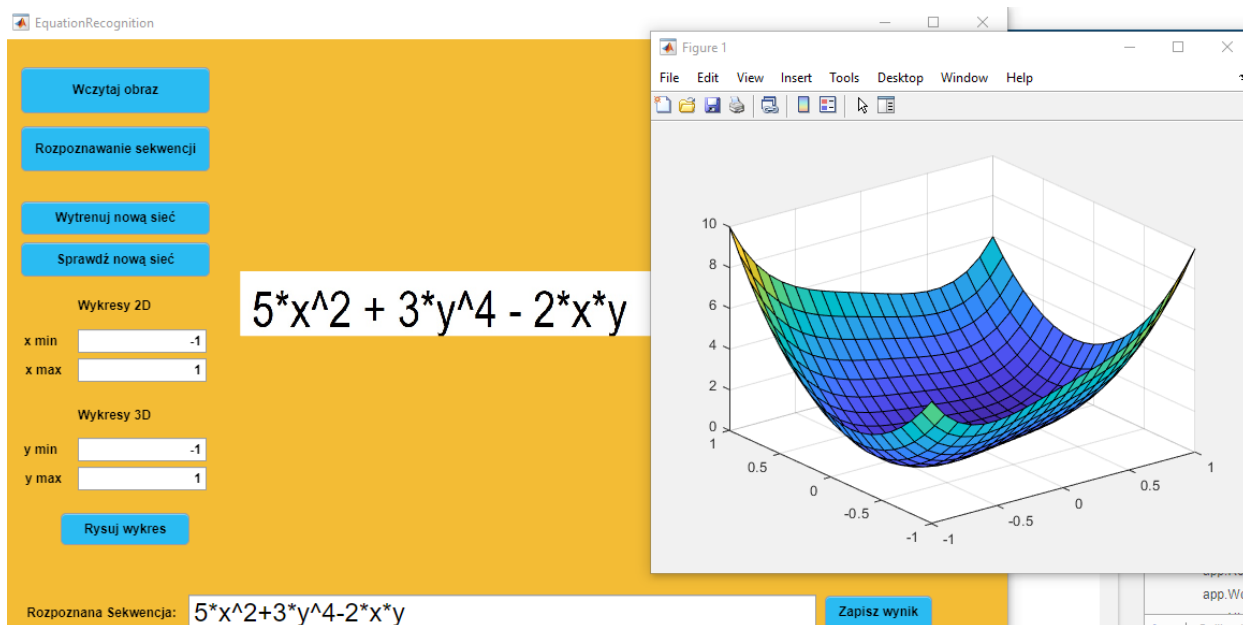
Rozpoznany tekst pojawia się w dolnej części ekranu opisanej jako output. Dodatkowo program oblicza wartość wpisanego wyrażenia gdy jest to możliwe



Rysunek 3: Ekran po kliknięciu przycisku "Rozpoznawanie tekstu"

Kolejny przycisk „Trenuj nową sieć” uruchamia proces uczenia nowej sieci. Następujący po nim przycisk „Sprawdź nową sieć” rozpoznaje równanie przy użyciu nowego zestawu współczynników z nowej sieci, przeciwnie do przycisku „Rozpoznawanie tekstu”, który używał starego, wcześniejszego zestawu obliczonych współczynników.

Analogicznie - rozpoznane działanie zostanie wyświetlony w polu tekstowym „Output”.



Rysunek 4: Ekran przy rysowaniu wykresu z rozpoznanego równania

Kolejne elementy interfejsu to Wykres 2D oraz Wykres 3D. Pozwalają one na rysowanie wykresu na podstawie odczytanego wyrażenia. Wymiar wykresu zależy od ilości zmiennych w rozpoznanym wyrażeniu. W zależności od potrzeb istnieje możliwość dostosowania przedziałów rysowanego wykresu

## 4. Wady projektu

- Program przestaje prawidłowo działać dla większej ilości zmiennych niż „x” i „y”, nie rozpoznaje innych zaawansowanych symboli matematycznych, złożonych znaków, znaków interpunkcyjnych lub odwróconego równania. Większość

zaproprowanych zmian mających na celu likwidację powyższych problemów kończyła się Validation Check'iem.

- Program nie potrafi rozpoznawać pisma odręcznego ani bardzo odmiennych stylów pisma maszynowego.

## 5. Przykładowe usprawnienia

- W przyszłości można by było rozbudować program o większą ilość zmiennych.
- Z racji, że na ten moment program rozpoznaje jedynie podstawowe działania matematyczne dobrym usprawnieniem byłoby wprowadzenie rozpoznawania pozostałych symboli matematycznych.

## 6. Podział obowiązków

**Implementacja sieci neuronowej, dokumentacja-** wspólnie

**generowanie danych,** - Maciej Walczak

**trenowanie sieci** - Łukasz Wajda

**tworzenie GUI** - Mateusz Niepokój