Spis treści

[WSTĘP 2](#_Toc496440407)

[ROZDZIAŁ PIERWSZY – CEL I ZAKRES PROJEKTU 3](#_Toc496440408)

[1.1 Cel projektu 3](#_Toc496440409)

[1.2 Zakres projektu 3](#_Toc496440410)

[ROZDZIAŁ DRUGI – OPIS DZIAŁANIA SIECI NEURONOWEJ 4](#_Toc496440411)

[ROZDZIAŁ TRZECI – IMPLEMENTACJA SYSTEMU WEBOWEGO 5](#_Toc496440412)

[3.1 Funkcjonalności systemu 5](#_Toc496440413)

[3.2 Baza Danych 5](#_Toc496440414)

[3.2 Środowisko Testowe 6](#_Toc496440415)

[ROZDZIAŁ CZWARTY – IMPLEMENTACJA SIECI NEURONOWEJ 7](#_Toc496440416)

[4.1 Biblioteka Keras 7](#_Toc496440417)

[4.2 Wczytywanie i obróbka obrazów wejściowych 7](#_Toc496440418)

[4.3 Zbieranie danych treningowych 7](#_Toc496440419)

[ROZDZIAŁ PIĄTY – TESTY I ANALIZA 9](#_Toc496440420)

[Bibliografia 10](#_Toc496440421)

# WSTĘP

Rozwój techniki dwudziestego pierwszego wieku, wraz ze stale rosnącymi możliwościami obliczeniowymi komputerów przyniosły przełom w dziedzinie uczenia maszynowego. Zadania, niegdyś możliwe do wykonania tylko na superkomputerach instytucji naukowych i potężnych firm dziś bez problemu wykonywane są za pomocą siły komputerów stacjonarnych, laptopów, a niedługo także i urządzenia mobilne zaczną sobie radzić z coraz to trudniejszymi zadaniami. Jednym z popularniejszych kierunków badań stały się sieci neuronowe.

„Sieć neuronowa jest bardzo uproszczonym modelem mózgu. Składa się ona z dużej liczby (od kilkuset do kilkudziesięciu tysięcy) elementów przetwarzających informacje”. Sieci neuronowe zdobyły swoją popularność dzięki dokładności połączonej z szybkością działania i zostaną szczegółowo opisane w rozdziale X.

Mimo swojej popularności trudno jest znaleźć system webowy, wykorzystujący algorytmy maszynowego uczenia, zwłaszcza jeśli chodzi o rozpoznawanie znaków w czasie rzeczywistym (z opóźnieniem, które nie zostanie zauważone przez użytkownika). Istniejące projekty, takie jak na przykład ten na stronie www. deepart.io potrzebują wiele czasu na kalkulacje i przetworzenie obrazów wejściowych. Zaimplementowany w tej pracy system postara się to zmienić dzięki zastosowaniu danych wejściowych o małym rozmiarze, w dodatku mocno uproszczonych. Dokładność stawiana będzie na drugim miejscu względem szybkości działania.

Tak więc praca ta ma za zadanie zintegrowanie szybkiej i względnie dobrze nauczonej sieci neuronowej, z systemem webowym w taki sposób, by użytkownik nie odczuł dyskomfortu płynącego z opóźnienia spowodowanego predykcją. Składać się ona(praca) będzie z Y Rozdziałów, wstępu, oraz podsumowania.

# ROZDZIAŁ PIERWSZY – CEL I ZAKRES PROJEKTU

## 1.1 Cel projektu

Celem niniejszego projektu jest opracowanie systemu webowego, którego głównym elementem będzie środowisko testujące responsywność zaimplementowanej sieci neuronowej, w formie minigry.

## 1.2 Zakres projektu

Praca składać będzie się z trzech głównych części, oraz dwóch modułów pomocniczych.

Pierwszą, a zarazem największą częścią będzie system webowy. Oferować on będzie możliwość tworzenia oraz logowania klientów na własne konto, przeglądania swoich danych oraz statystyk, a także uruchomienia minigry/środowiska testowego, które omówione zostanie w akapicie poniżej.

Drugą częścią będzie środowisko testowe, dzięki któremu możliwe będzie zbadanie responsywności oraz jakości modelu sieci neuronowej. Zostanie ono zaimplementowane w formie minigry, w której zadaniem gracza będzie narysowanie za pomocą ruchu myszy określonego znaku, który następnie będzie przekazywany do algorytmu neuralnego.

Trzecia część, to wspomniana wcześniej siec neuronowa. Zaimplementowana w języku Python, z użyciem biblioteki Keras będzie miała za zadanie najpierw nauczyć się rozpoznawania znaków wejściowych, a następnie rozpoznawania narysowanych przez gracza w czasie rzeczywistym.

Moduły pomocnicze, o których wspomniano w pierwszym akapicie tego rozdziału, to:

- Zaimplementowany w języku c++ z użyciem biblioteki sfml program pozwalający na rysowanie, przechwytywanie oraz zapisywanie obrazów na twardy dysk. Będzie on niezwykle pomocny, przy zbieraniu danych testowych potrzebnych do uczenia sieci neuronowej.

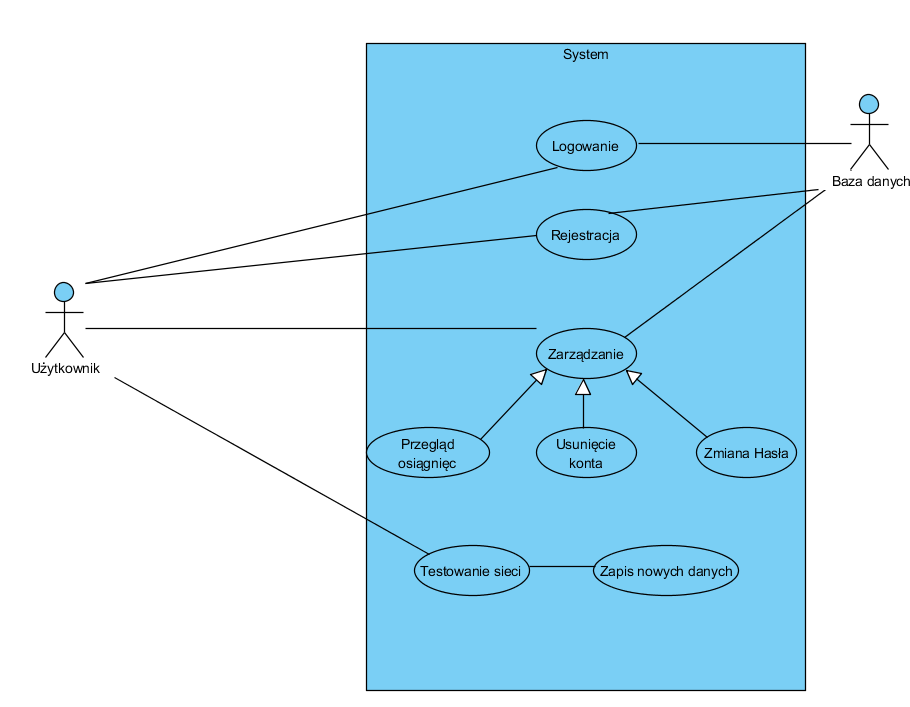
- Baza danych wykonana w języku MySql.

# ROZDZIAŁ DRUGI – HISTORIA I DZIAŁANIE SIECI NEURONOWEJ

# ROZDZIAŁ TRZECI – IMPLEMENTACJA SYSTEMU WEBOWEGO

## 3.1 Funkcjonalności systemu

Wszystkie możliwe funkcje systemu, z punktu widzenia użytkownika przedstawione zostały na diagramie przypadków użycia (Rysunek 3.1.1).



Rysunek 3.1.1 – Diagram przypadków użycia systemu webowego

Jak widać na powyższym diagramie, użytkownik będzie miał możliwość Rejestracji w systemie webowym, która to czynność, jeśli zostanie doprowadzona do końca wywoła utworzenie nowego wpisu w bazie danych.

Jeśli konto zostało już utworzone możliwe będzie zalogowanie do systemu, gdzie po wpisaniu przez użytkownika danych – nazwy użytkownika oraz hasła – wysłane zostanie zapytanie do bazy danych, czy konto takiego użytkownika istnieje i czy hasło jest prawidłowe.

Po zalogowaniu, zostanie udostępniona użytkownikowi możliwość zarządzania swoimi danymi. Do wyboru będą trzy opcje:

- zmiana hasła – po podaniu obecnego hasła i potwierdzeniu jego poprawności przez bazę danych, możliwość zaaktualizowania hasła na inne.

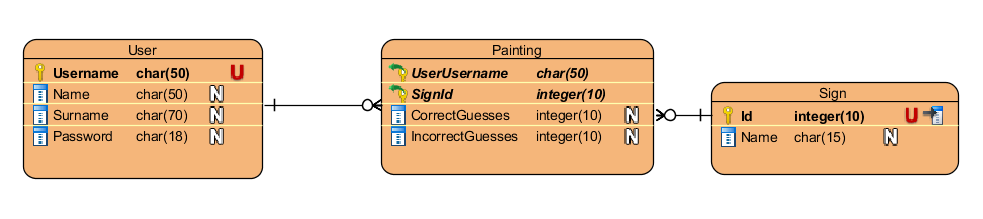
- Usunięcie konta – jeśli użytkownik podejmie decyzję o zakończeniu korzystania z usług serwisu, będzie miał możliwość usunięcia swojego konta, a co za tym idzie wszystkich swoich danych osobowych z bazy danych. W bazie pozostaną jednak dane dotyczące narysowanych znaków, zmienione tak, aby nie dało się ich przypisać usuniętemu kontu. W celu usunięcia konta niezbędne będzie podanie hasła, które zostanie zatwierdzone przez Bazę danych.

- Przegląd osiągnięć – użytkownikowi udostępniony zostanie panel, dzięki któremu będzie mógł sprawdzić własne statystyki dotyczące narysowanych wcześniej znaków, z podziałem na te, które system rozpoznał i te, które rozpoznane nie zostały.

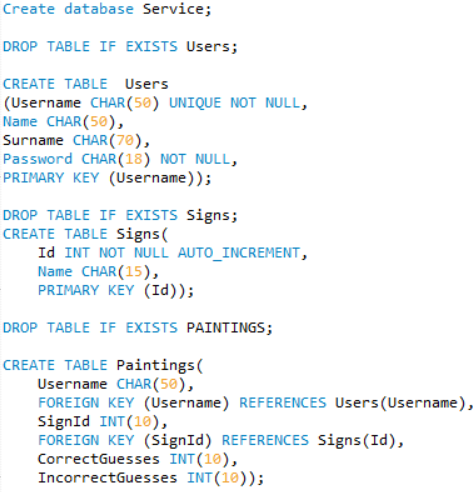
Ostatnią opisaną opcją jest uruchomienie środowiska testowego, gdzie użytkownik ma możliwość rysowania znaków, oraz oceny odpowiedzi sieci neuronowej (dobrze rozpoznane lub źle rozpoznane).

## 3.2 Baza Danych

W celu zapewnienia możliwości przechowywania danych użytkowników, a także zapisu danych z testów – ilość poprawnie oraz niepoprawnie rozpoznanych przez sieć znaków rysowanych przez użytkownika - przygotowano bazę danych w języku MySql. Baza składa się z trzech encji – jednej dla użytkownika, drugiej, w której przechowywane są dane każdego znaku, oraz encji zliczającej poprawne oraz niepoprawne narysowanie każdego rodzaju znaku, przez każdego z osobna użytkownika. Strukturę bazy danych przedstawia diagram na rysunku 3.2.1, natomiast skrypt, za pomocą którego zaimplementowano bazę przedstawiono na rysunku 3.2.2 .



Rysunek 3.2.1 – Struktura bazy danych serwisu



Rysunek 3.2.2 – Skrypt tworzący bazę danych

W przyszłości, jeśli system będzie nadal rozwijany baza danych będzie mogła zostać łatwo poszerzona o nowe encje, czy atrybuty.

## 3.2 Środowisko Testowe

# ROZDZIAŁ CZWARTY – IMPLEMENTACJA SIECI NEURONOWEJ

## 4.1 Biblioteka Keras

## 4.2 Wczytywanie i obróbka obrazów wejściowych

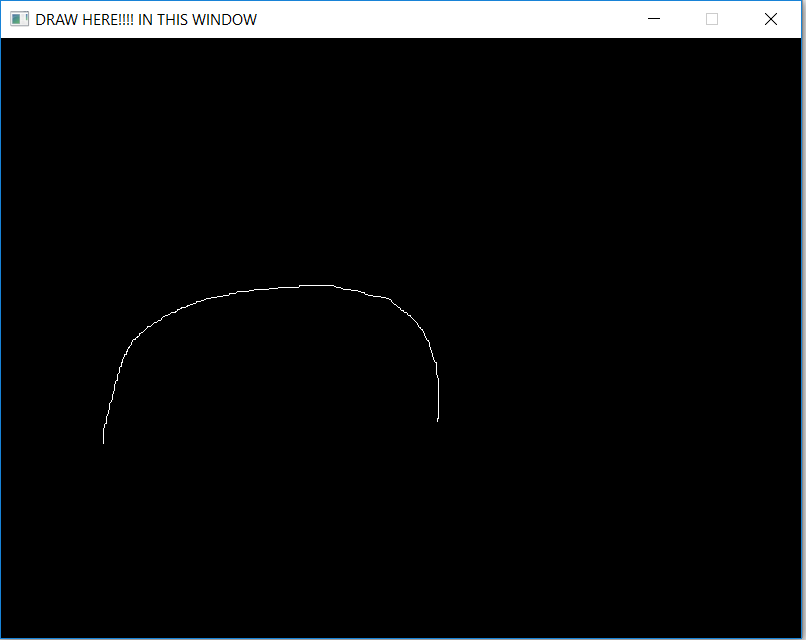
## 4.3 Zbieranie danych treningowych

Na rysunkach od 4.3.1 do 4.3.9 przedstawiono pierwowzory obrazów, których rozpoznawania uczyć się będzie sieć neuronowa. Dla osiągnięcia zadowalających rezultatów nauczania sieci neuronowej zapewnić należy odpowiednią ilość danych uczących, czyli w tym wypadku kilku tysięcy „ręcznie”(za pomocą myszy) narysowanych znaków. Przyspieszenie procesu zbierania znaków, a także możliwość skorzystania z pomocy innych osób umożliwi program pomocniczy, pozwalający na rysowanie na ekranie obrazu. Program został wykonany w języku c++ z użyciem biblioteki SFML (biblioteka pozwala między innymi na przechwycenie pozycji myszy na ekranie oraz zapisanie danych do obrazu .png).

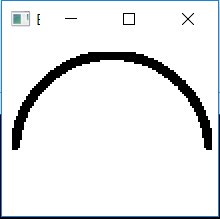
Działanie programu jest banalnie proste – tworzone są na ekranie komputera dwa okna (rysunek 4.3.10 oraz 4.3.11), jedno pozwala na rysowanie znaku, drugie pokazuje wzorzec. Program, po wciśnięciu prawego przycisku myszy zaczyna przechwytywać pozycję myszy w oknie, i zapisuje ją do odpowiedniej struktury. Kolejne naciśnięcie przycisku powoduje konwersję zebranych danych do pliku .png, losowanie kolejnego wzorca oraz wyczyszczenie struktur czytających pozycję myszy. Dodatkowo uniemożliwiono użytkownikom zmianę wielkości okna rysowania, co pozwala na zapewnienie, że każdy z zapisanych obrazów będzie tej samej wielkości. Funkcjonalność tego programu jest bardzo podobna do środowiska testowego umieszczonego w systemie będącym przedmiotem niniejszej pracy, jednak pozbawiony zbędnych obliczeń związanych z obecnością sieci neuronowej lepiej nadaje się do zapisu wielu plików jednocześnie. Na rysunkach 4.3.10 oraz 4.3.11 przedstawiono wygląd programu.

|  |  |  |
| --- | --- | --- |
| C:\Users\Master\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Image1.png | C:\Users\Master\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Image2.png | C:\Users\Master\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Image3.png |
| C:\Users\Master\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Image4.png | C:\Users\Master\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Image5.png | C:\Users\Master\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Image6.png |
| C:\Users\Master\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Image7.png | C:\Users\Master\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Image8.png | C:\Users\Master\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Image9.png |

Rysunki 4.3.1 -4.3.9 – Wzory obrazów dla sieci neuronowej



Rysunek 4.3.10 – Okno rysowania



Rysunek 4.3.11 – okno z przykładowym znakiem

# ROZDZIAŁ PIĄTY – TESTY I ANALIZA

# Bibliografia

**Brak źródeł w bieżącym dokumencie.**