Analiza Obrazów
Projekt
Rozpoznawanie tekstu

Julia Bała
Tomasz Madej
Dawid Wojdylak

27 stycznia 2022

Spis treści

1 Opis realizowanego tematu

Temat realizowanego przez nas projektu to transkrypcja na zapis tekstowy tekstu drukowanego z wczytywanego przez użytkownika obrazu.

Przy wstępnych założeniach aplikacja miała odseparowywać od siebie kolejne znaki (litery oraz cyfry) oraz je rozpoznawać. Wyodrębnienia poszczególnych znaków dokonaliśmy przy użyciu operacji morfologicznych, natomiast za pomocą technologii uczenia maszynowego nauczyliśmy program rozpoznawać te znaki.

2 Instrukcja instalacji programu i uruchomienia

W celu instalacji programu wymagany jest Python3 oraz pip. Z poziomu głównego katalogu instalację przeprowadzamy następująco:

```
1        $ pip install .
2        $ export PATH="/home/<NAZWA\_UZYTKOWNIKA>/.local/bin:\$PATH\$"
3        $ text-reader
```

Alternatywnie można

```
python3 setup.py
```

lub uruchomić z folderu src

```
python3 main.py
```

lub

```
./main.py
```

(ta opcja wymaga ręcznej instalacji pakietów).

3 Instrukcja obsługi programu

Po uruchomieniu programu pojawia się główne okno. Podstawowe menu stanowi górny pasek z opcjami Exit; Load; Image; Save Text; Reset; Train Model.

Aby wczytać obraz wybieramy przycisk Load Image. Otworzy się okno wyboru pliku. W projekcie zawarty jest folder *tmp_ data*, a w nim przykładowe obrazy.

Następnym krokiem jest wybór wyuczonego modelu za pomocą przycisku Load Model. Ponownie otwiera się okno wyboru pliku. Wybieramy folder *model* oraz odpowiedni plik.

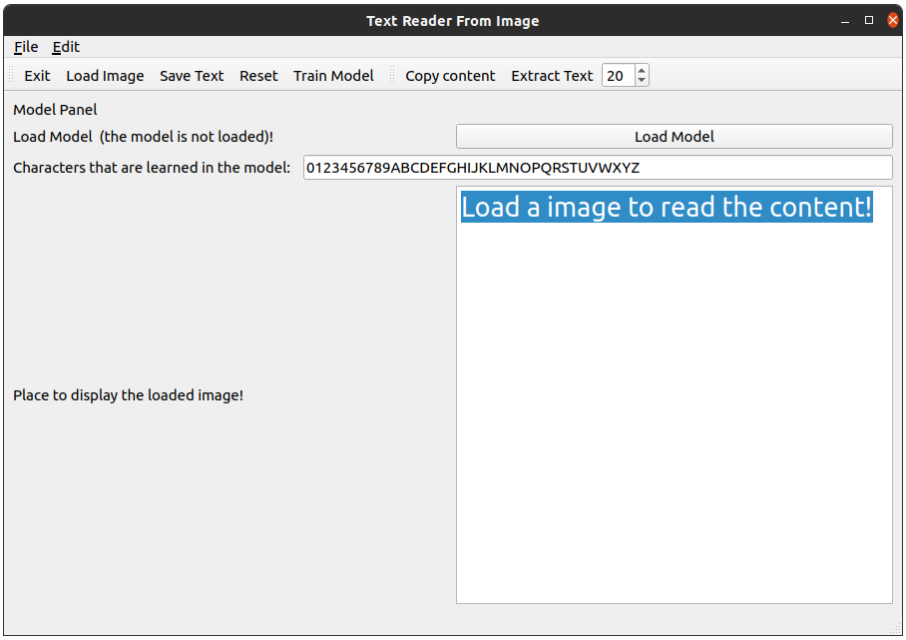
Z dostępnych wyszkolonych przez nas modeli dostępne są:

- CapitalLettersAndNumbersBlackChars.h5
- modelNum[1...9]whitedigitals.h5

Pierwszy służy do rozpoznawania wielkich czarnych liter oraz cyfr na białym tle, drugi do rozpoznawania białych cyfr na czarnym tle. Do każdego obrazu należy odpowiednio dobrać model. W dodatku pole **Characters that are learned in the model** musi odpowiadać wyuczonym znakom.

Istnieje możliwość wyszkolenia dodatkowych modeli przy użyciu okna Train Model, co jest opisane w sekcji **5.**

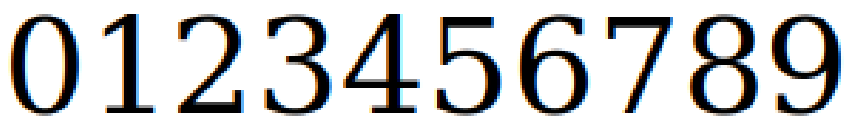
Po kliknięciu w przycisk Extract Text wykonują się odpowiednie algorytmy odpowiedzialne za wyodrębnienie znaków oraz ich rozpoznanie. Po ukończeniu tego procesu rozpoznany tekst pokazuje się w prawym polu aplikacji. Wielkość czcionki możemy dostrajać ostatnim polem w głównym menu.



Rysunek 1: Okno główne programu

4 Przykładowe dane

Poniżej przestawiono przykładowe zdjęcia, na których przeprowadzono testy aplikacji. Zgodnie z instrukcją obsługi programu, należy wybrać odpowiedni model rozpoznawania tekstu. Testy przebiegły pomyślnie, z wyjątkiem Rysunku 5: cyfra 8 mylona jest z 0. Wynika to najprawdopodobniej z niewystarczającego przeszkolenia modelu.



Rysunek 2: Przykładowy plik *num2.png*

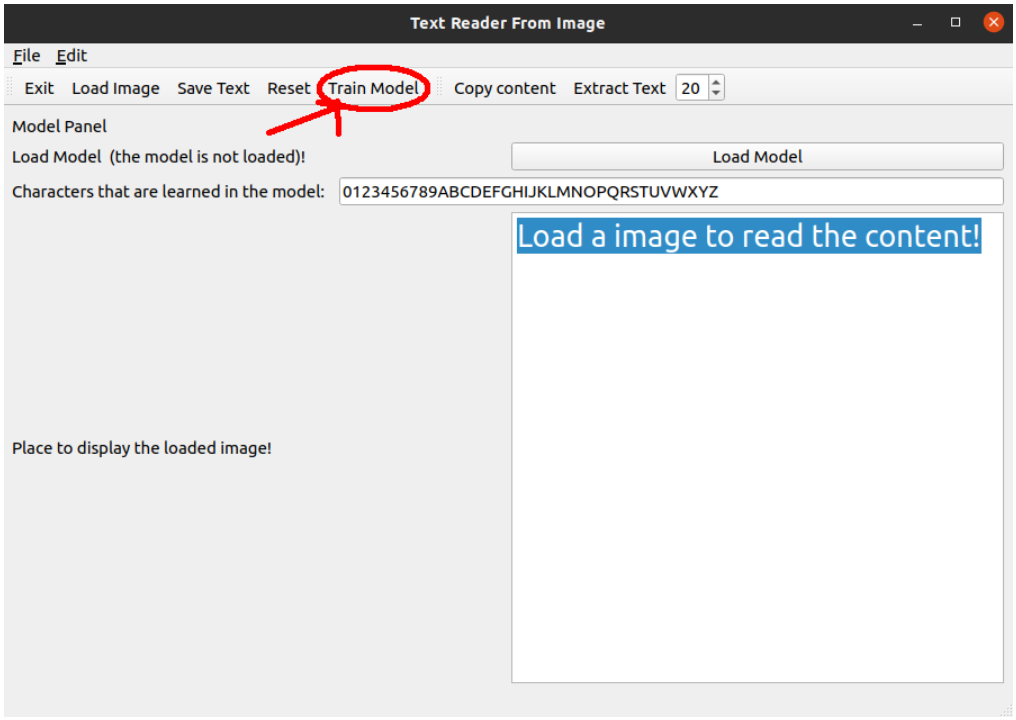
Rysunek 3: Przykładowy plik *RO_.png*

Rysunek 4: Przykładowy plik *testSerif.png*

Rysunek 5: Przykładowy plik *tmpNum.png*

5 Uczenie maszynowe

Aby wyszkolić model do odczytywania znaków należy przygotować zdjęcia znaków nas literujących i umieścić je w folderach training i validation, a dla każdego znaku utworzyć folder z nazwą tego znaku. W aplikacji wchodzi w okno uczenia maszynowego przedstawionego na Rysunek 6.

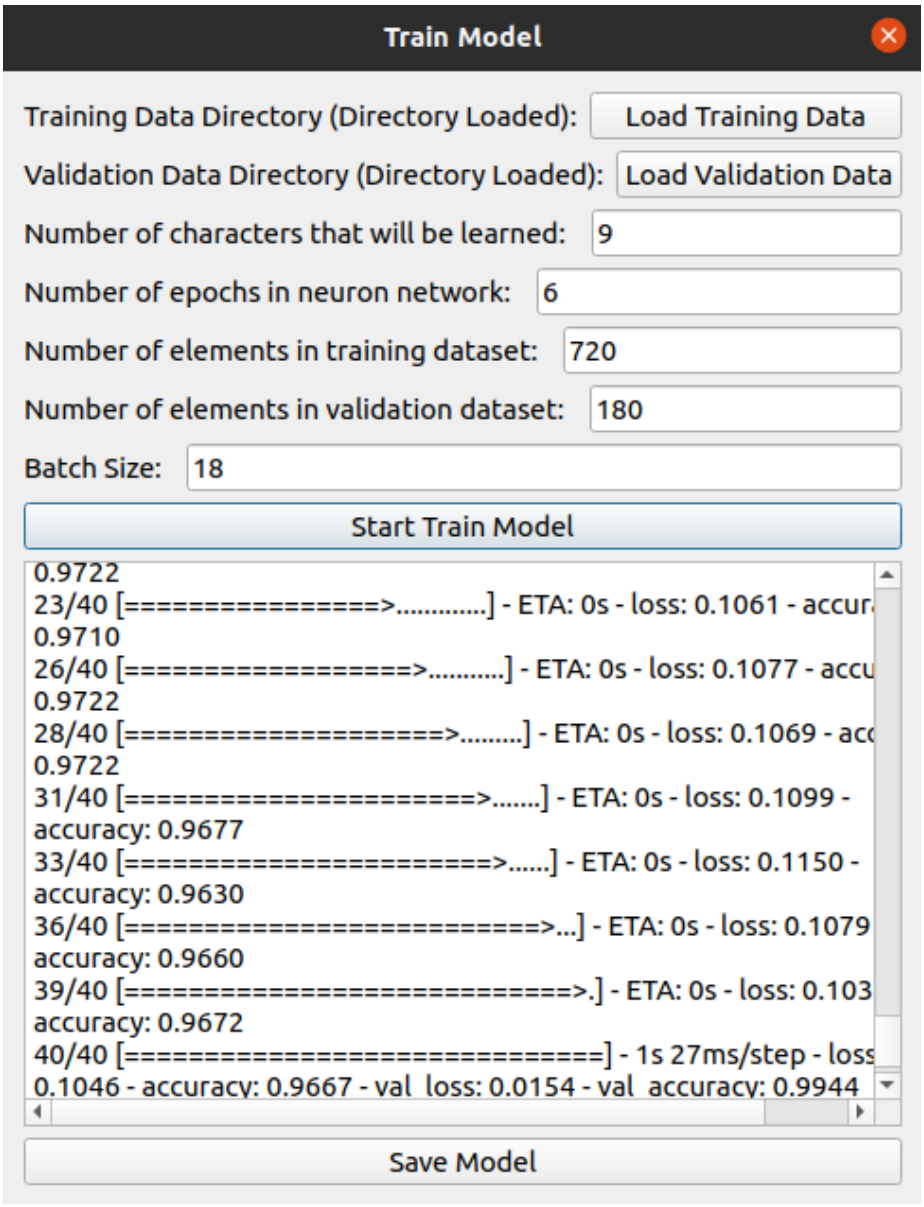


Rysunek 6: Wskazanie przycisku otwierającego panel uczenia maszynowego.

W odtworzonym oknie należy podać następujące dane:

- wczytać dane do uczenia/walidacji,
- suma znaków jaki się będzie uczyć,
- ilość epok uczenia maszynowego,
- łączna ilość elementów w folderze training and validation,
- batch size - wpływa na ilość elementów które będą propagowane

Po wprowadzeniu tych danych można rozpocząć uczenie, które chwilkę może potrwać w zależności od tego jak dużą bazę przygotowaliśmy. Po skończonym algorytmie należy zapisać model. Przykładowe uczenie można zobaczyć na Rysunek 7.



Rysunek 7: Przykładowy wynik przeprowadzonego uczenia.

6 Co nie działa?

- Program działa jedynie dla pojedynczych linii tekstu składających się z wielkich liter oraz cyfr

- Odstępy pomiędzy słowami nie są rozpoznawane (po transkrypcji tekstu powstaje zwarty ciąg znaków)
- Małe litery są błędnie rozpoznawane jako inne znaki.
- Pomiędzy poszczególnymi znakami w słowie musi zostać zachowany odpowiedni odstęp, aby zostały poprawnie odseparowane

Przedstawione powyżej niedoskonałości aplikacji występują dla wyszkolonego przez naszych modelu. Projekt umożliwia wyszkolenie kolejnych modeli na podstawie większej ilości danych. I

7 Podział obowiązków

- opracowanie algorytmu uczenia maszynowego
  uczenie modelu
  dokumentacja
  Tomasz Madej
- opracowanie algorytmu separacji poszczególnych znaków
  dodanie przykładowych danych
  dokumentacja
  Dawid Wojdylak
- GUI
  testowanie danych
  dokumentacja
  Julia Bała