

# Aplikacja odczytująca wyrazy z obrazów

Jagoda Kuczkowska

# Cel projektu

Celem projektu było stworzenie systemu rozpoznawania tekstu na obrazach (OCR) z wykorzystaniem własnego modelu CRNN.

Chciałam, aby model był w stanie automatycznie odczytywać napisy z obrazków, co znajduje zastosowanie np. w digitalizacji dokumentów, automatycznym odczycie tablic rejestracyjnych czy przetwarzaniu zdjęć z tekstem



# Zbiór danych

Do nauki i testowania modelu wykorzystałam zbiór MJSynth. Jest to duży, syntetyczny zbiór danych, który zawiera miliony obrazków przedstawiających napisy w różnych czcionkach, stylach i zniekształceniach.

W projekcie używałam obrazków w skali szarości o rozmiarze 128x32 piksele.

Przed treningiem obrazki były skalowane i normalizowane, aby model mógł się szybciej i skuteczniej uczyć.





# Architektura modelu


Model CRNN (Convolutional Recurrent Neural Network) łączy w sobie dwie główne części:

01.

Warstwy konwolucyjne (CNN): wyciągają z obrazka najważniejsze cechy, takie jak kształty liter, linie czy łuki.

02.

Warstwy rekurencyjne (LSTM): analizują sekwencję tych cech, aby rozpoznać kolejność liter w napisie. Na końcu model ma warstwę softmax, która dla każdej „kolumny” obrazka przewiduje, jaka litera tam występuje.



# Przygotowanie danych

01

Konwersja do skali szarości

02

Skalowanie

obrazki były zmieniane na rozmiar 128x32 piksele.

03

Normalizacja

wartości pikseli były dzielone przez 255, aby mieściły się w zakresie 0-1

04

Tokenizacja

napisy były zamieniane na ciągi liczb odpowiadających literom w słowniku.

05

Padding

napisy były uzupełniane do tej samej długości, aby można było je batchować



Batch size:  
20



Liczba epok:  
100



Optymalizator:  
Adam

# Proces uczenia

Callbacki:

- EarlyStopping – przerywa trening, jeśli nie ma poprawy.
- ModelCheckpoint – zapisuje najlepszy model.
- ReduceLROnPlateau – zmniejsza learning rate, jeśli model się „zatrzyma”.



Monitorowanie postępu:

- Używałam TensorBoard do wizualizacji procesu uczenia.
- Po treningu rysowałam wykresy strat i dokładności.





# Ewaluacja

Po treningu model był oceniany na zbiorze testowym.  
Sprawdzałam:

- Dokładność na poziomie znaków: ile pojedynczych liter model rozpoznał poprawnie.
- Dokładność na poziomie słów: ile całych napisów model odczytał bezbłędnie.
- Edit distance: ile zmian (dodanie, usunięcie, zamiana liter) potrzeba, by poprawić przewidziany napis do prawidłowego. Wyniki pokazały, że model świetnie sobie radzi. ale czasem myli podobne litery lub gubi się przy dłuższych słowach.

appear

WINNESSES

STALKER

SHUCKSES

NATIVES



The background is a light gray color, decorated with various hand-drawn blue doodles. These include several overlapping circles and ovals at the top, a wavy line at the bottom center, and several checkmarks at the bottom right. There are also some abstract scribbles and lines scattered throughout the corners.

**Dziękuję za  
uwagę**