

Софийски университет "Св. Климент Охридски"

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

Специалност "Извличане на информация и откриване на знания"

## CyrillicOCR

Разпознаване на ръкописен текст на кирилица



Изготвил: Даниел Халачев

Факултетен номер: 4MI3400603

Преподавател: проф. Иван Койчев

Дата: 22 юни 2025 г.

# Съдържание

<b>1 Увод</b>	<b>3</b>
1.1 Мотивация . . . . .	3
<b>2 Преглед на областта</b>	<b>3</b>
2.1 Съществуващи решения . . . . .	4
2.1.1 CRNN . . . . .	4
2.1.2 Seq2Seq (2019) . . . . .	4
2.1.3 Vision Transformer (ViT) (2021) . . . . .	4
2.1.4 TrOCR (2021) . . . . .	5
<b>3 Архитектура на системата</b>	<b>5</b>
<b>4 Реализация</b>	<b>6</b>
4.1 Имплементация . . . . .	6
4.2 Трениране на модела . . . . .	6
4.2.1 Набори от данни . . . . .	7
4.2.2 Хиперпараметри . . . . .	7
4.3 Резултати . . . . .	9
4.3.1 Метрики . . . . .	9
4.3.2 Трениране . . . . .	9
4.3.3 Резултати върху тестовото множество . . . . .	10
<b>5 Заключение</b>	<b>11</b>
<b>A Полезни набори от данни за разпознаване на текст на кирилица</b>	<b>12</b>
A.1 Kaggle . . . . .	12
A.2 HuggingFace . . . . .	12

## **Декларация за липса на плагиатство**

- Плагиатство е да използваш, идеи, мнение или работа на друг, като претендиш, че са твои. Това е форма на преписване.
- Тази курсова работа е моя, като всички изречения, илюстрации и програми от други хора са изрично цитирани.
- Тази курсова работа или нейна версия не са представени в друг университет или друга учебна институция.
- Разбирам, че, ако се установи плагиатство в работата ми, ще получа оценка “Слаб”.

**Даниел Иванов Халачев**

# 1 Увод

## 1.1 Мотивация

България, както и целият свят, се намират в процес на дигитална революция, който включва изоставянето на традиционни носители в полза на електронното съхраняване на информация. То позволява значително по-бърза агрегация, обработка и търсене в данните в сравнение с традиционни носители като хартията.

Но дигитализацията в България (и не само) изостава. Множество документи все още се попълват на ръка, а стари записи на информация все още не са дигитализирани. В допълнение, множество статии и книги, особено по-малко популярни, все още не са налични в дигитален формат.

Откриването на текст в изображения (*text recognition*) е добре изучен проблем, за който съществуват решения с висока точност. Но те са адаптирани основно към латинската азбука. В областта все още липсват значими изследвания и експерименти в разпознаването на текст на кирилица, особено за ръкописен текст.

Този курсов проект цели да предложи архитектура за дълбоко самообучение, подходяща за разпознаване на ръкописен текст на кирилица.

## 2 Преглед на областта

Съществуват две основни интерпретации на проблема по извлечане на текст от изображения:

- **Character Recognition** - поединично разпознаване на отделни символи
- **Text Recognition** - разпознаване на думи и дори цели изречения наведнъж.

За разпознаването на пасажи от текст е по-подходящ вторият похват. Причината е, че разпознатите символи при *Character Recognition* трябва да бъдат групирани в думи. Откриването къде започва и свършва думата е нова задача сама по себе си.

Разпознаването на текст с похватта *Text Recognition* използва значително по-обемисти и сложни модели, но позволява разпознаването на цели думи и изречения наведнъж.

Друга разбивка на проблема е по това какъв текст се разпознава:

- печатан текст
- ръкописен текст
- и двата вида

Разпознаването на ръкописен текст е по-сложно от разпознаването на печатан текст поради две основни причини:

- тренировъчните набори за печатан текст са значително повече. В допълнение, могат лесно да бъдат генерирали синтетични тренировъчни данни с помощта на компютърните шрифтове
- ръкописните букви имат значително повече детайли и вариациите в изписването на буквите са значително по-големи.

В допълнение, съществува и поддисциплина *разпознаване на текст в сцени* - специализиране в разпознаването на текст в обръщажаващата среда (напр. табели и надписи в паркове, улици, сгради и др.).

## 2.1 Съществуващи решения

### 2.1.1 CRNN

Архитектурата *CRNN (2015)* комбинира няколко вида невронни мрежи в себе си<sup>[9]</sup>:

- **Convolutional Neural Network (CNN)** - конволюционната невронна мрежа извлича характеристики на буквите
- **Recurrent Neural Network (RNN)** - рекурентната невронна мрежа обработва последователностите от извлечени характеристики, като за разпознаването на един символ помага контекста - съседните символи
- **Connectionist Temporal Classification (CTC)** - превръща вероятностите за срещане на символите в самите символи, като взема предвид последователността на появата им в текста.

### 2.1.2 Seq2Seq (2019)

*Sequence-to-Sequence (Seq2Seq)* комбинира конволюционна невронна мрежа като енкодер на характеристиките с декодер, базиран на внимание<sup>[5]</sup>. Архитектурата първоначално е разработена за откриване на текст в сцени. Методът значително подобрява разпознаването на изкривен и деформиран текст с помощта на механизмите за внимание, насочени към конкретни части от изображението.

### 2.1.3 Vision Transformer (ViT) (2021)

*Vision Transformer (ViT)* моделът ViTSTR е първият от това семейство за разпознаване на текст<sup>[1]</sup>. Използва се за разпознаване на текст в сцени, като обработва изображенията като последователности от ивици. Използва внимание към себе си (self-attention), за да прихване глобални зависимости. В резултат се постига точност,

съпоставима с най-добрите модели по това време, но със значително по-малко параметри, в сравнение със CNN-базираните модели. Тази архитектура бележи началото на използването на *Transformer* модели за разпознаване на текст, като се използват претренирани модели за ефективност.

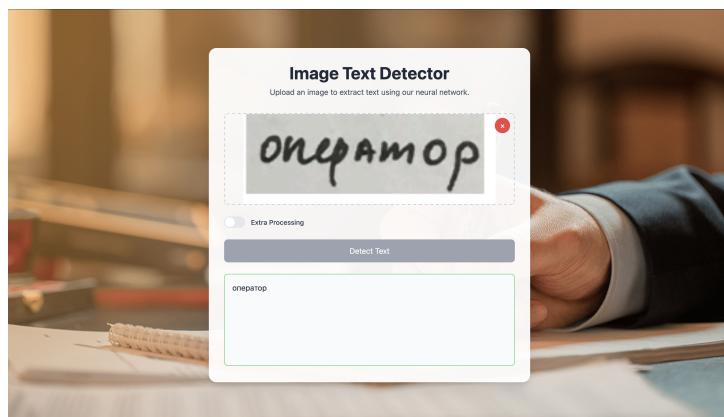
#### 2.1.4 TrOCR (2021)

TrOCR използва два претренирани *Transformer* модела за разпознаване на текст<sup>[6]</sup>. Първият е претрениран Vision Transformer, ползван като енкодер, а вторият - претрениран *Transformer* за текст (*RoBERTa*) в ролята на декодер. Моделът постига рекордни резултати в разпознаването на печатан и ръкописен текст и текст от сцени и е най-добрият в областта към момента.

### 3 Архитектура на системата

Архитектурата на невронната е вдъхновена както от *CNN*, така и от *Transformer* моделите. Тя се състои от два компонента:

- **Backbone**: Извлича характеристиките от изображенията. Може да бъде два варианта:
  - **CNN**: Конволюционна невронна мрежа.
  - **DeformableCNN**: Деформируема конволюционна невронна мрежа<sup>[2]</sup>.
- **OCRModel**: Обединение на първия компонент с *Transformer* модел. Последният приема изхода на конволюционната мрежа и се обучава да съпоставя характеристиките на поредици от символи.
- **OCRModelWrapper**: Обвива целия модел заедно с конфигурационни класове, за по-лесно предаване във функциите за трениране. Имплементира функция **inference** за експлоатация на вече тренирания модел.
- графичен интерфейс



Фигура 1: Потребителски интерфейс на Cyrillic-OCR

## 4 Реализация

### 4.1 Имплементация

Невронната мрежа е имплементирана на *Pytorch*. Имплементацията използва съществуваща разработка<sup>[10]</sup> като отправна точка за смислени стойности по подразбиране на хиперпараметрите на трансформъра, но я надгражда с модифициран `Backbone` модул и различен подбор и аугментация на тренировъчните набори от данни. Проектът използва `uv` за управление на виртуалната среда и зависимостите. Платформата *Weights & Biases* се използва за логване на временните и крайните данни от тренирането на модела.

За изработването на графичния интерфейс са използвани *Flask*, *HTML*, *CSS*, *JS* и *Tailwind CSS*.

В допълнение към основните модули за архитектурата, дефинирани в секция 3, са реализирани още няколко помощни класа:

- `NaturalDataloader`: специален клас за зареждане на естествения набор данни *Cyrillic Handwriting Dataset*<sup>[11]</sup>.
- `SyntheticDataloader`: специален клас за зареждане на синтетичния набор от данни *Synthetic Cyrillic Large Dataset*<sup>[8]</sup>.
- `OCRModelConfig`: конфигурационен файл за модела, който включва:
  - азбука и специални символи
  - тип на `Backbone`.
  - брой скрити слоеве на трансформъра
  - брой кодиращи и декодиращи слоеве на трансформъра
  - брой глави на трансформъра
  - вероятност за *dropout*
  - размери на входното изображение
  - средно и стандартно отклонение на наборите от данни, след като са преработени в черно-бялата гама
- прост графичен интерфейс, който позволява лесно взаимодействие с модела.

Структурата на проекта следва следната йерархия:

### 4.2 Трениране на модела

Невронната мрежа беше тренирана на нает за целта сървър през платформата <https://vast.ai>. Графична карта на сървъра е `Nvidia H100 80GB HBM3`.

Epoch	Image	Ground Truth	Prediction	CER	WER
36		светской	светской	0	0
36		элементов.	элементов.	0	0
37		начала	начала	0	0
37		истор.	истор.	0	0
37		Ласпейреса	Ластей	0.5	5
37		в арктическую зону	в аритическую	0.3333	2

Фигура 2: Резултати по време на тренирането на епохи 36 и 37.

#### 4.2.1 Набори от данни

За тренирането бяха използвани следните набори от данни:

Име	Размер
pumb-ai/cyrillic-handwritten-large <sup>[8]</sup>	3,800,000†
constantinwerner/cyrillic-handwriting-dataset <sup>[11]</sup>	73,830

Таблица 1: Набори от данни за трениране. †Поради ограничения във времевите и финансовите ресурси, само 25% от синтетичния набор от данни бяха използвани за тренирането.

И двата набора от данни съдържат само *train* и *test* дялове. Затова с цел генерирането на *validation* дял тренировъчният дял бе разделен на две части в отношение 95:5.

#### 4.2.2 Хиперпараметри

Поради ограничените времеви и финансови ресурси, беше проведен само един експеримент със следните хиперпараметри:

Хиперпараметър	Описание	Стойност
<code>backbone_type</code>	Вид ResNet на <code>Backbone</code>	<code>RESNET_50</code>
<code>hidden</code>	Скрити слоеве на трансформъра	512
<code>enc_layers</code>	Енкодер слоеве на трансформъра	5
<code>dec_layers</code>	Декодер слоеве на трансформъра	4
<code>nhead</code>	Глави на трансформъра	8
<code>dropout</code>	Вероятност за <i>dropout</i>	0.1
<code>width</code>	Ширина на входното изображение	256
<code>height</code>	Височина на входното изображение	64
<code>max_length</code>	Максимална продължителност на изхода	100
<code>natural_mean</code> <code>synthetic_mean</code>	Средно на естествения набор Средно на синтетичния набор	$\begin{bmatrix} 0.75645548 \\ 0.75645548 \\ 0.75645548 \end{bmatrix}$
<code>natural_std</code> <code>synthetic_std</code>	Дисперсия на естествения набор Дисперсия на синтетичния набор	$\begin{bmatrix} 0.23744543 \\ 0.23744543 \\ 0.23744543 \end{bmatrix}$
<code>synthetic_batch_size</code>	Размер на партидата (сint. набор)	128
<code>natural_batch_size</code>	Размер на партидата (ест. набор)	128
<code>synthetic_lr</code>	Скорост на обучение (pretraining)	$2e^{-4}$
<code>natural_lr</code>	Скорост на обучение (fine-tuning)	$2e^{-4}$
<code>synthetic_decay_rate</code>	Степен на отслабване (pretraining)	$1e^{-2}$
<code>natural_decay_rate</code>	Степен на отслабване (fine-tuning)	$1e^{-2}$
<code>synthetic_epochs</code>	Максимален брой епохи (pretraining)	10
<code>natural_epochs</code>	Максимален брой епохи (fine-tuning)	50
<code>patience</code>	Максимален брой епохи без подобреие	5

Таблица 2: Хиперпараметри на модела

## 4.3 Резултати

### 4.3.1 Метрики

За трениране на модела се използва критерий *Cross-Entropy Loss*.

За оценка на модела се прилагат следните метрики:

- **Characted Error Rate (CER)** - брой сгрешени (подменени  $S_c$ , излишни  $I_c$  или липсващи  $D_c$ ) символи спрямо общия брой. Различия в малки и главни букви не се отчитат.

$$\text{CER} = \frac{S_c + I_c + D_c}{N_c}$$

- **Word Error Rate (WER)** - брой сгрешени (подменени  $S_w$ , излишни  $I_w$  или липсващи  $D_w$ ) думи спрямо общия брой. За грешка се счита и излишни или липсващи интервали между думите. Различия в малки и главни букви не се отчитат.

$$\text{WER} = \frac{S_w + I_w + D_w}{N_w}$$

### 4.3.2 Трениране

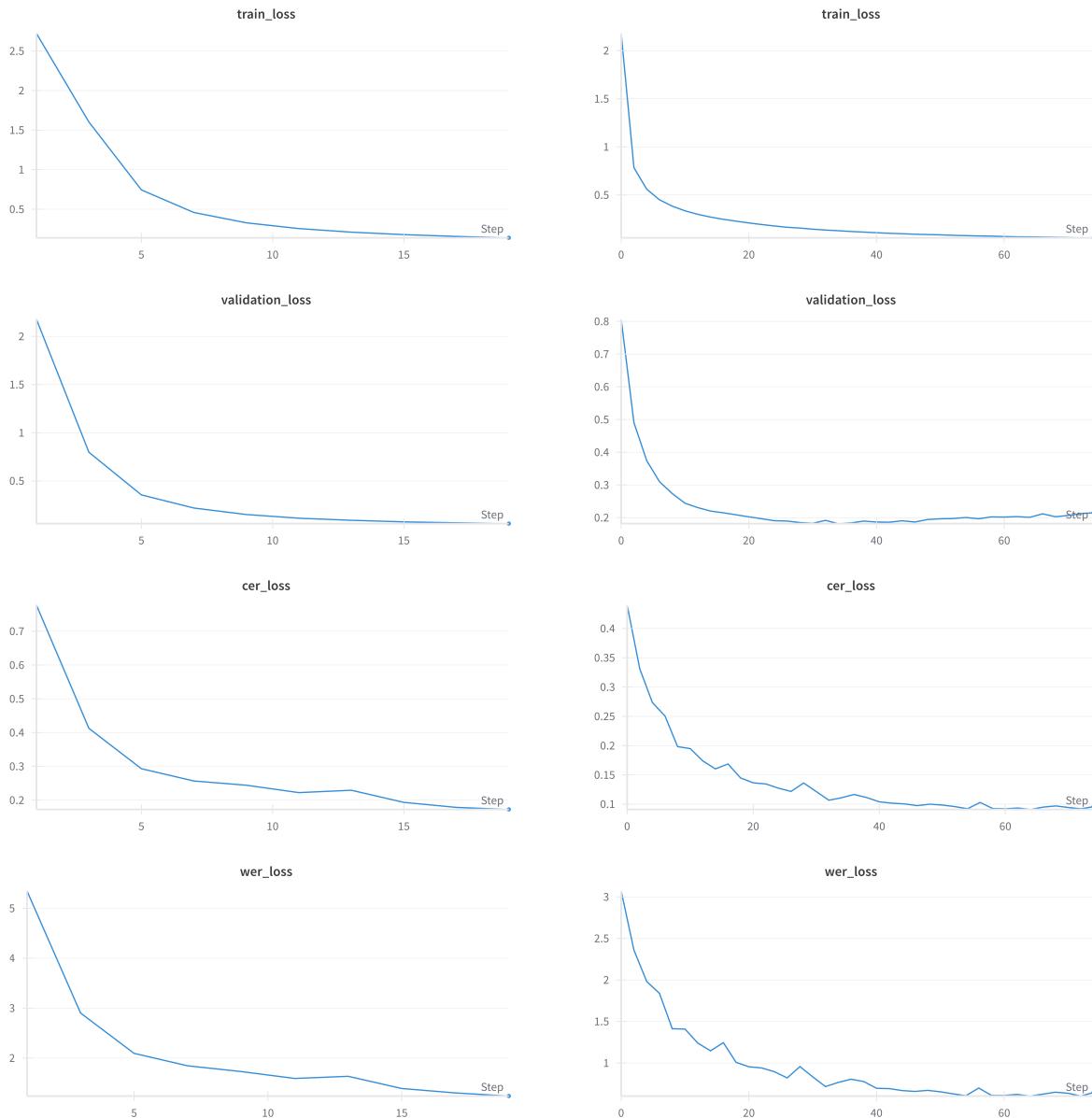
Предтрениране за 10 епохи върху 25% от синтетичния набор от данни и трениране за 37 епохи върху естествения набор от данни след 5 епохи без подобреие.

Metric	Value
Training loss	0.1384
Validation loss	0.0570
Validation CER	0.1784
Validation WER	1.2408
Време за трениране	80 mins

Таблица 3: Резултати от предтренирането

Metric	Value
Training loss	0.0546
Validation loss	0.2148
Validation CER	0.0958
Validation WER	0.6449
Време за трениране	25 mins

Таблица 4: Резултати от допълнително трениране



Фигура 3: Метрики по време на предтrenирането

Фигура 4: Метрики по време на допълнителното трениране

#### 4.3.3 Резултати върху тестовото множество

Метрика	Стойност
Test CER	0.1590
Test WER	1.1698

Таблица 5: Резултати над тестовото множество

## 5 Заключение

Проектът демонстрира ефективността на комбинацията от *CNN* и *Transformer* за разпознаване на ръкописен текст на кирилица.

Въпреки ограничените времеви и финансови ресурси и един единствен експеримент със стойностите на хиперпараметрите, моделът постига добра точност върху всички набори от данни. Резултатите върху тестовото множество са по-ниски от тези на валидационното множество, което е често срещан ефект, предизвикан от разделението на набора от данни.

Съществуват възможности за надграждане на проекта чрез експериментиране с различни комбинации на хиперпараметри, пълно използване на синтетичния набор от данни, използване на модула с деформируема конволюция **DeformableCNN** вместо стандартна конволюция, както и трениране с допълнителни набори от данни<sup>[3;4;7]</sup>.

## **A Полезни набори от данни за разпознаване на текст на кирилица**

### **A.1 Kaggle**

- Cyrillic Handwriting Dataset:  
<https://www.kaggle.com/datasets/constantinwerner/cyrillic-handwriting-dataset>
- PPT OCR Data of 8 Languages:  
<https://www.kaggle.com/datasets/nexdatafrank/ppt-ocr-data-of-8-languages>
- Optical Character Recognition Dataset:  
<https://www.kaggle.com/datasets/tamirpuzanov/nto-task2-dataset>
- RusTitW:  
[www.kaggle.com/datasets/hardtype/rustitw-russian-language-visual-text-recognition](https://www.kaggle.com/datasets/hardtype/rustitw-russian-language-visual-text-recognition)
- OCR Receipt Text Detection:  
<https://www.kaggle.com/datasets/trainingdatapro/ocr-receipts-text-detection>

### **A.2 HuggingFace**

- DonkeySmall:  
[https://huggingface.co/DonkeySmall.](https://huggingface.co/DonkeySmall)
- GAN (synthetic) Cyrillic:  
[https://huggingface.co/datasets/nastyboget/gan\\_cyrillic](https://huggingface.co/datasets/nastyboget/gan_cyrillic)
- Stackmix Cyrillic Large:  
[https://huggingface.co/datasets/nastyboget/stackmix\\_cyrillic\\_large](https://huggingface.co/datasets/nastyboget/stackmix_cyrillic_large)
- Synthetic Cyrillic Large:  
[https://huggingface.co/datasets/pumb-ai/synthetic-cyrillic-large.](https://huggingface.co/datasets/pumb-ai/synthetic-cyrillic-large)
- Synthdog Multilingual:  
<https://huggingface.co/datasets/WueNLP/Synthdog-Multilingual-100>
- HWR200:  
<https://huggingface.co/datasets/AntiplagiatCompany/HWR200>
- School Notebook Dataset:  
[https://huggingface.co/datasets/ai-forever/school\\_notebooks\\_RU](https://huggingface.co/datasets/ai-forever/school_notebooks_RU)

## Литература

- [1] Rowel Atienza. Vision transformer for fast and efficient scene text recognition. <https://arxiv.org/abs/2105.08582>, 2021.
- [2] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. <https://arxiv.org/abs/1703.06211>, 2017.
- [3] Evgenii Davydkin, Aleksandr Markelov, Egor Iuldashev, Anton Dudkin, and Ivan Krivorotov. Data generation for post-ocr correction of cyrillic handwriting. <https://arxiv.org/abs/2311.15896>, 2023.
- [4] Sharon Fogel, Hadar Averbuch-Elor, Sarel Cohen, Shai Mazor, and Roee Litman. Scrabblegan: Semi-supervised varying length handwritten text generation. <https://arxiv.org/abs/2003.10557>, 2020.
- [5] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. <https://arxiv.org/abs/1811.00751>, 2019.
- [6] Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. Trocr: Transformer-based optical character recognition with pre-trained models. <https://arxiv.org/abs/2109.10282>, 2022.
- [7] Naphat Nithisopa and Teerapong Panboonyuen. Dota: Deformable optimized transformer architecture for end-to-end text recognition with retrieval-augmented generation. <https://arxiv.org/abs/2505.04175>, 2025.
- [8] pumb-ai. Synthetic Cyrillic large. <https://huggingface.co/datasets/pumb-ai/synthetic-cyrillic-large>, 2023.
- [9] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. <https://arxiv.org/abs/1507.05717>, 2015.
- [10] vergotten. CyrillicOCR-ResNet-Transformer. <https://github.com/vergotten/CyrillicOCR-ResNet-Transformer>, 2023.
- [11] Constantin Werner. Cyrillic Handwriting Dataset. <https://www.kaggle.com/datasets/constantinwerner/cyrillic-handwriting-dataset>, 2023.