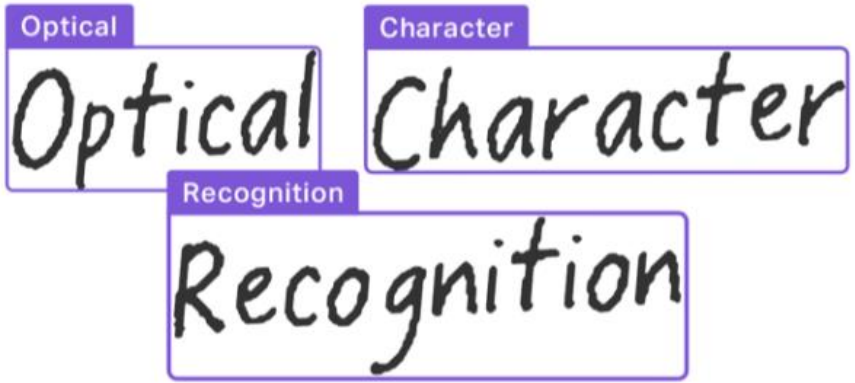
Распознавание рукописного текста

Введение

Оптическое распознавание символов (Optical character recognition, OCR) - это преобразование изображений, содержащих рукописный или печатный текст, в текст, воспринимаемый и интерпретируемый компьютером.



Подходы к распознаванию текста

- Классические подходы компьютерного зрения,
- Подходы с использованием нейронных сетей.

Классические CV подходы

Как правило, такой подход включает в себя следующие шаги:

1. Применение **фильтров** для выделения символов от фона,
2. **Нахождение областей**, на которых может располагаться текст,
3. **Выделение контуров** (границ) для распознавания отдельных символов,
4. **Классификация** каждого символа.

Нейросетевые подходы

Современное же ПО использует для распознавания текста нейронные сети, которые обучаются распознавать целые строки текста вместо того, чтобы фокусироваться на отдельных символах

Сложности распознавания рукописного текста

- Огромная изменчивость и неоднозначность штрихов от человека к человеку,
- Стиль почерка отдельного человека может меняться время от времени,
- Низкое качество исходного документа/изображения из-за ухудшения с течением времени,
- Текст в печатных документах располагается по прямой линии, в то время как люди могут писать строку текста не по прямой линии,
- Скорописный почерк затрудняет разделение и распознавание символов,
- Текст, написанный от руки, может иметь поворот вправо (влево), что отличается от печатного текста, где весь текст расположен прямо,
- Сбор хорошего размеченного набора данных для изучения обходится недешево по сравнению с синтетическими данными.

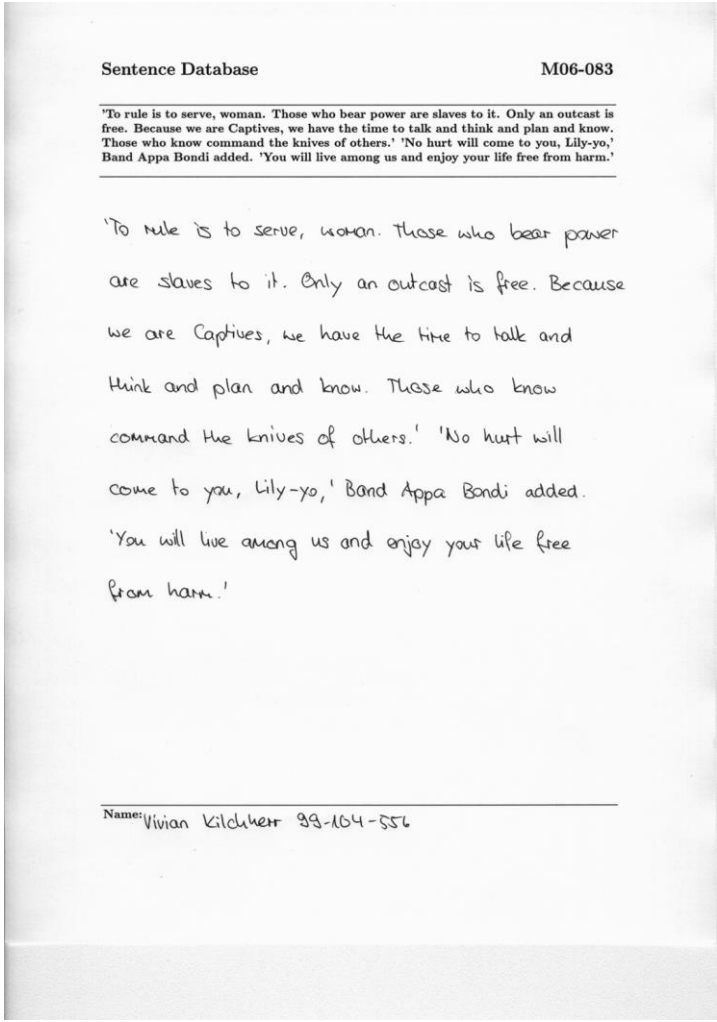
IAM dataset

Обширный набор данных рукописных текстов, в создании которых приняли участие 657 человек. Содержит:

- **1539** страниц текста,
- 5685 отдельных размеченных предложений,
- 13353 отдельных размеченных строк,
- 115320 отдельных размеченных слов.

Тексты были предварительно отсканированы и сохранены с разрешением 300 dpi, в формате PNG с 256 уровнями серого.

Одна из страниц датасета IAM



Sentence Database M06-083

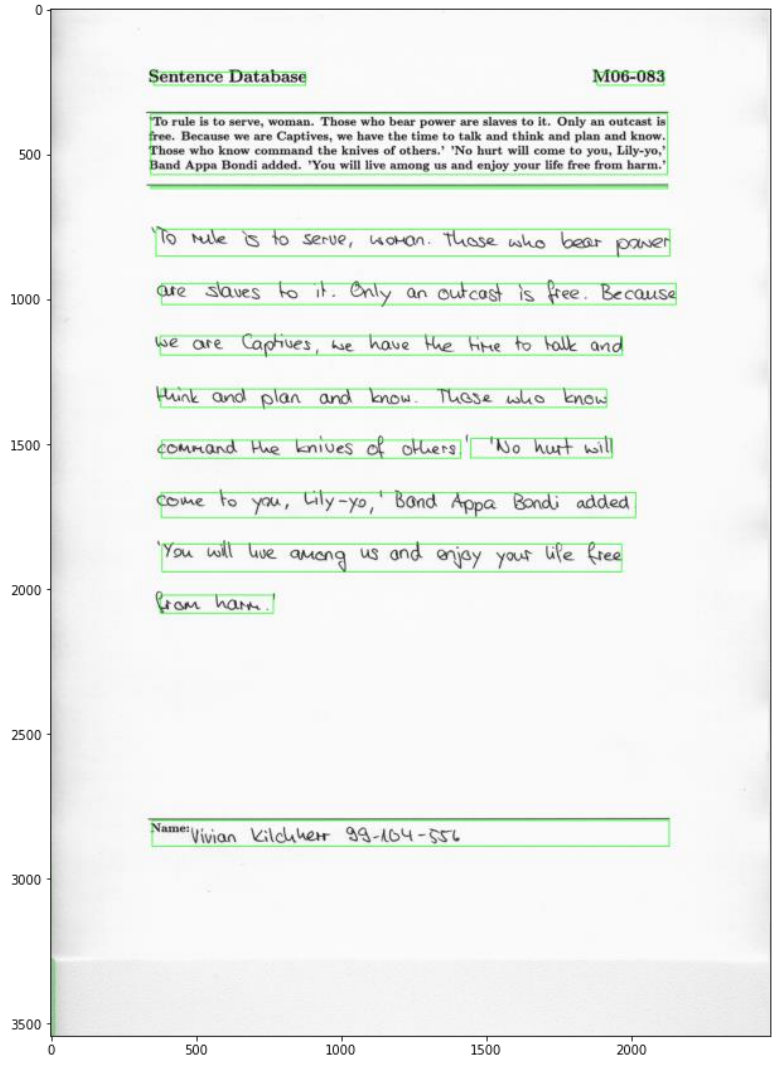
'To rule is to serve, woman. Those who bear power are slaves to it. Only an outcast is free. Because we are Captives, we have the time to talk and think and plan and know. Those who know command the knives of others.' 'No hurt will come to you, Lily-yo,' Band Appa Bondi added. 'You will live among us and enjoy your life free from harm.'

'To rule is to serve, woman. Those who bear power are slaves to it. Only an outcast is free. Because we are Captives, we have the time to talk and think and plan and know. Those who know command the knives of others.' 'No hurt will come to you, Lily-yo,' Band Appa Bondi added. 'You will live among us and enjoy your life free from harm.'

Name: Vivian Kilchherr 99-104-556

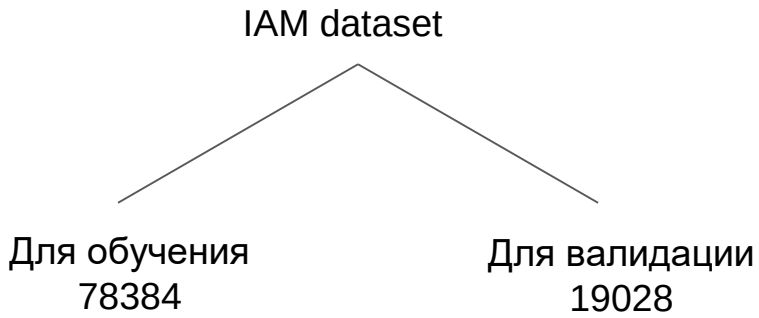
Выделенные области интереса (RoI)

1. Бинаризация изображения
2. Расширение (Dilation)
3. Сужение (Erosion)
4. Выделяем получившиеся прямоугольники

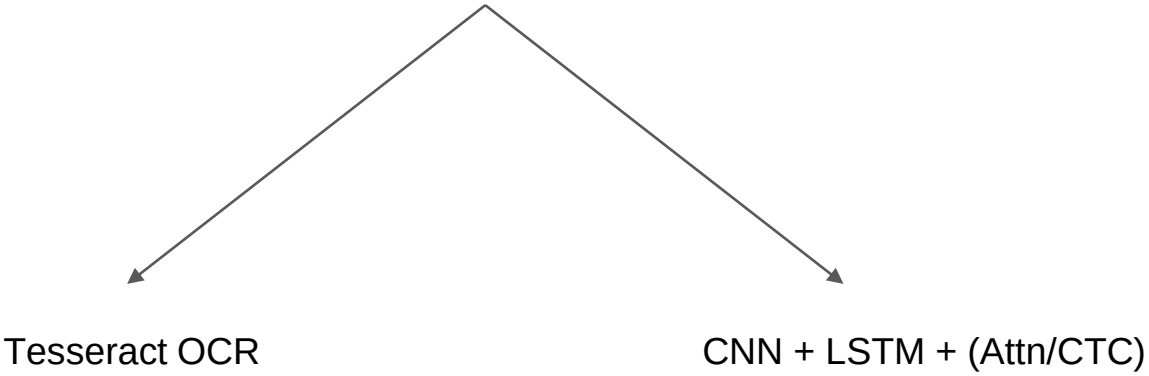


Распознавание рукописных слов

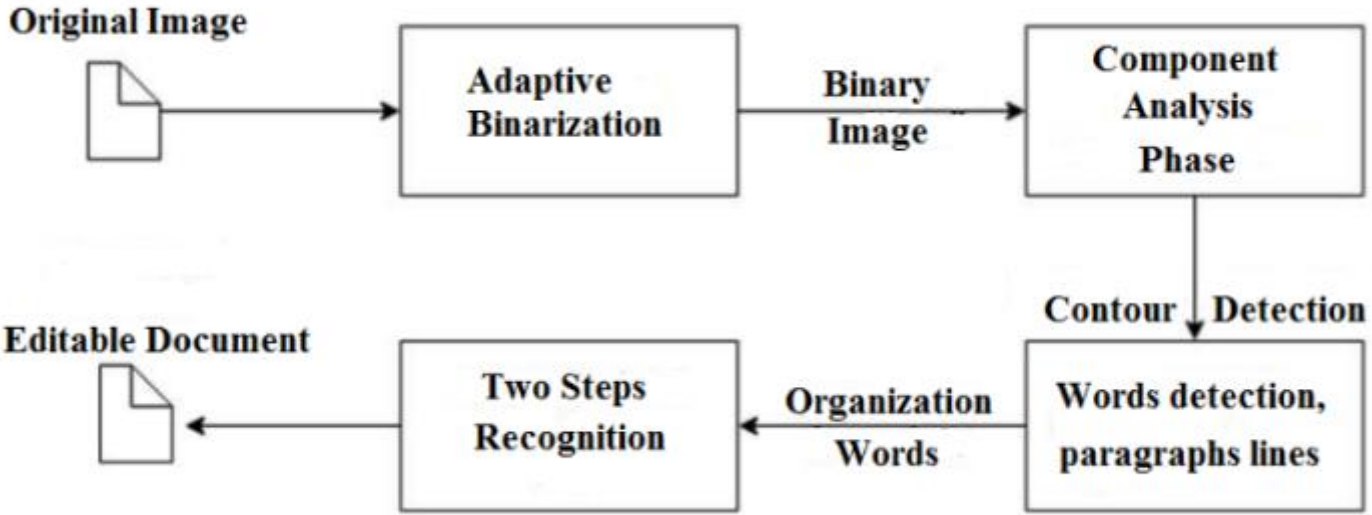
Склеенные в предложение слова из датасета IAM



Рассмотренные модели



Tesseract 3


Original Image
Adaptive Binarization
Binary Image
Component Analysis Phase
Contour Detection
Words detection, paragraphs lines
Organization Words
Two Steps Recognition
Editable Document

Tesseract 4, 2016 г.

LSTM + Static shape classifier

CNN + LSTM + Connectionist temporal classification (CTC)

Сверточная ResNet

Визуальные признаки

Рекуррентная BiLSTM

Контекстные признаки

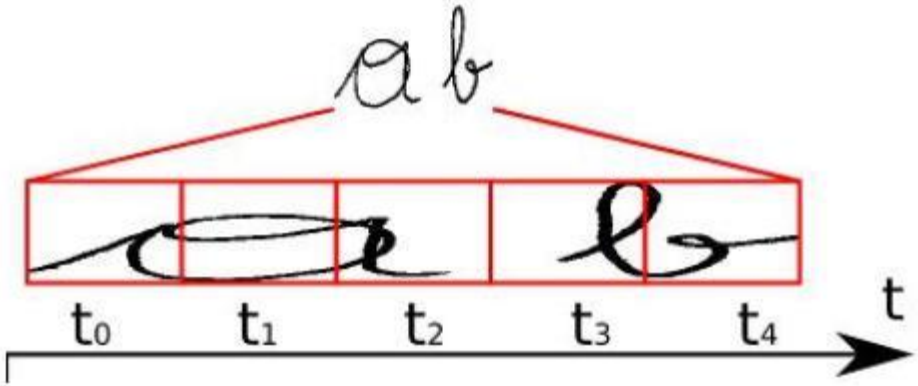
L-EE-A-RR-N-I-NN-G

CTC decoder

Предсказание

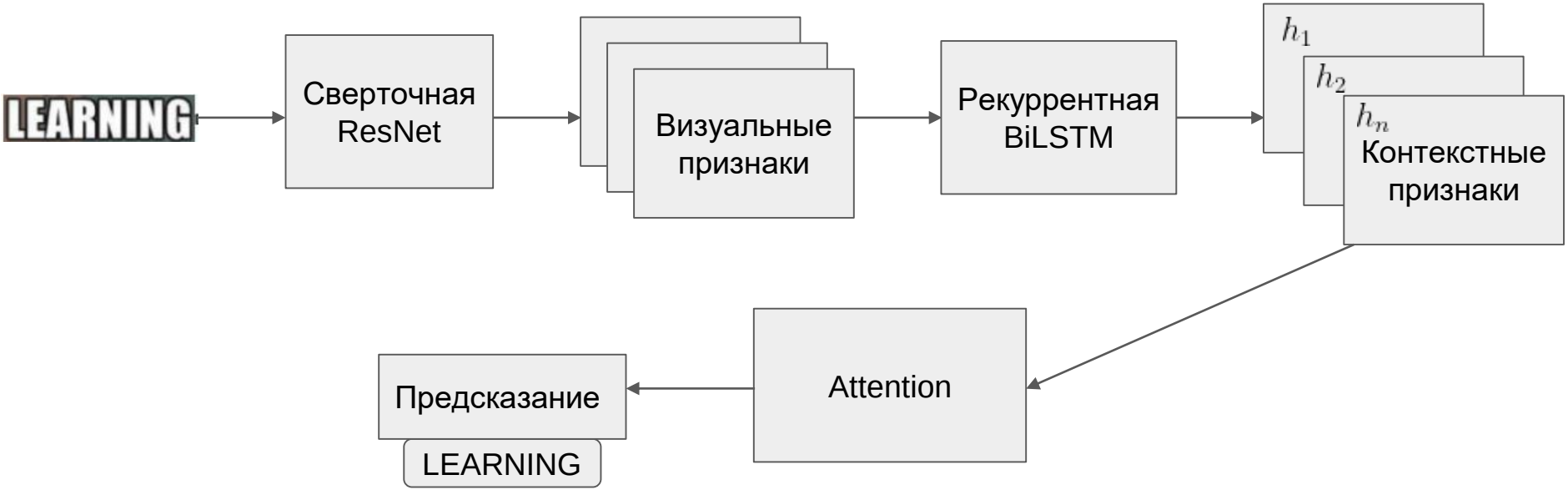
LEARNING

Connectionist temporal classification (CTC) декодер

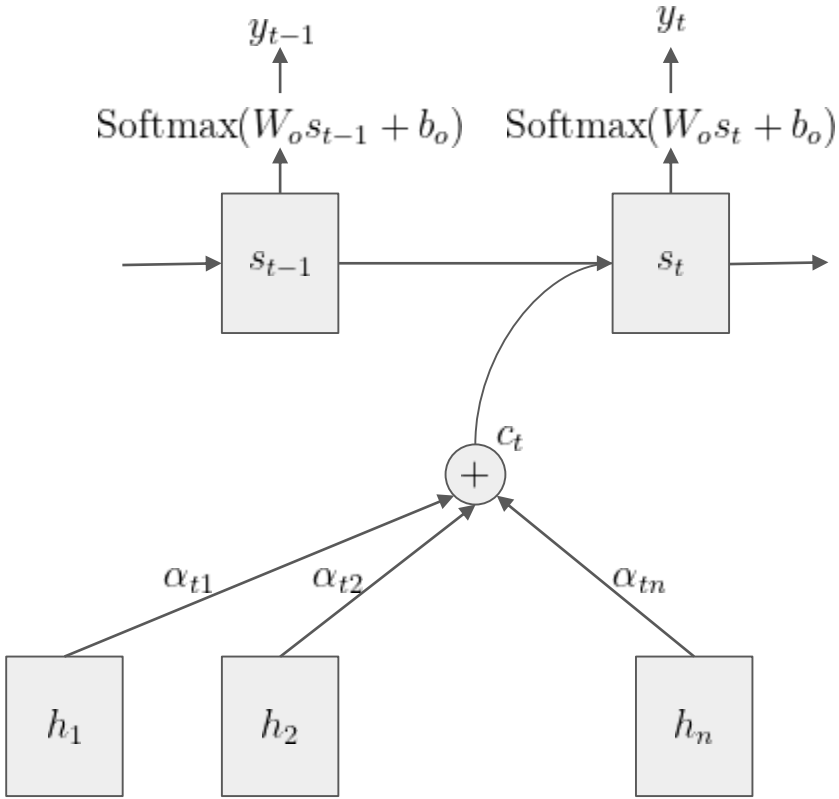


aaabb ⇒ ab

CNN + LSTM + Attention (Attn)



LSTM attention decoder



$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{n} \exp(e_{tk})}$$

$$e_{ti} = v^T \tanh(W s_{t-1} + V h_t + b)$$

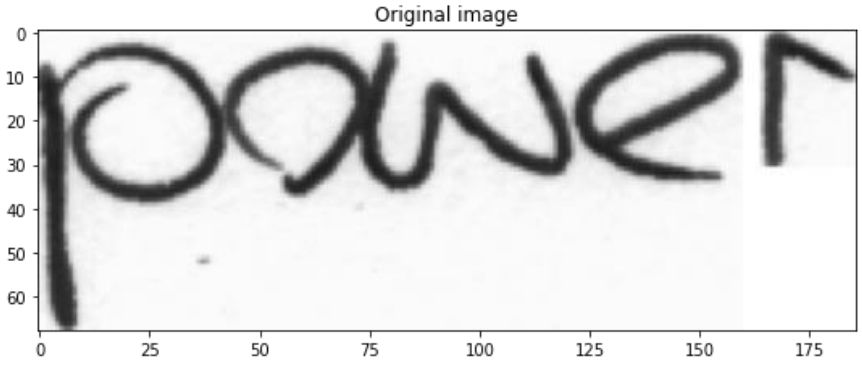
$$c_t = \sum_{i=1}^{n} \alpha_{ti} h_i$$

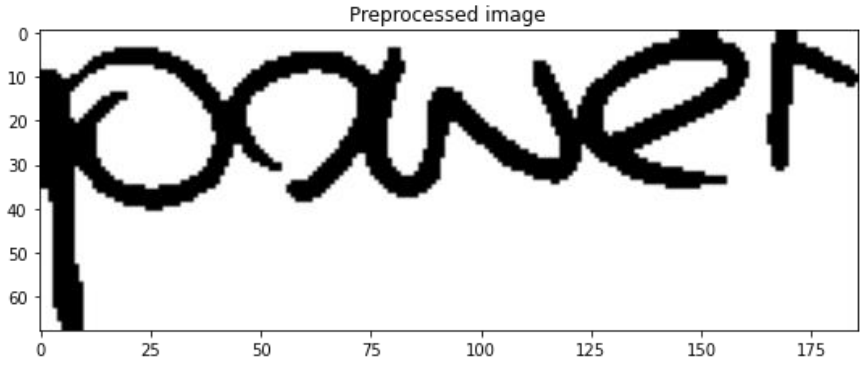
$$s_t = \text{LSTM}(y_{t-1}, c_t, s_{t-1})$$

$$y_t = \text{Softmax}(W_o s_t + b_o)$$

Предобработка изображений слов

- Из RGB в оттенки серого
- Небольшое размытие
- С помощью метода Оцу (Otsu’s method) приводим к черно-белому
- Применяем морфологическую операцию закрытия





Method		Accuracy	Norm-ED	Levenshtein	CER	Prediction Time, min	Training Time, min
Tesseract	with preprocessing, without fine-tuning	0.05	0.29	3.41	0.86	54	-
ResNet-BiLSTM-Attn	without preprocessing, without fine-tuning (1 iter.)	37.30	0.60	1.94	0.41	1.18	
	without preprocessing, with fine-tuning (500 iter.)	**81.76**	**0.93**	**0.34**	**0.08**	1.18	27.5
	with preprocessing, without fine-tuning (1 iter.)	35.33	0.58	2.04	0.48	1.18	
	with preprocessing, with fine-tuning (500 iter.)	81.08	0.92	0.36	0.09	1.18	**19.92**
ResNet-BiLSTM-CTC	without preprocessing, without fine-tuning (1 iter.)	31.52	0.63	1.59	0.40	1.18	
	without preprocessing, with fine-tuning (500 iter.)	76.64	0.91	0.41	0.09	1.18	27.38
	with preprocessing, without fine-tuning (1 iter.)	34.80	0.65	1.51	0.36	1.18	
	with preprocessing, with fine-tuning (500 iter.)	76.40	0.91	0.42	0.09	1.18	19.97

На чем обучались модели?

Tesseract, latin-based languages
400000 строк текста, включающих в себя более 4500 шрифтов

Трехэтапная модель CNN-RNN-decoder

1) MJSynth (MJ) - синтетический датасет, содержит 9 млн. изображений на которых 90 тыс. слов разным шрифтом.
2) SynthText (ST) - синтетический датасет, содержит 800 тыс. изображений и ~8 млн. слов

MJSynth & SynthText

(a) MJSynth word boxes

(b) SynthText scene image

Метрики

- Accuracy
- Levenshtein

$$D(i,j) = \begin{cases} 0, & i = 0,\ j = 0 \\ i, & j = 0,\ i > 0 \\ j, & i = 0,\ j > 0 \\ \min\{ & \\ \quad D(i, j-1) + 1, & \\ \quad D(i-1, j) + 1, & j > 0,\ i > 0 \\ \quad D(i-1, j-1) + \mathrm{m}(S_1[i], S_2[j]) & \\ \} & \end{cases},$$

где $\mathrm{m}(a,b)$ равна нулю, если $a = b$ и единице в противном случае

Пример: “rain” -> “sain” -> “shin” -> “shine” (Расстояние Левенштейна = 3)

Метрики

- Norm-ED (нормированный Levenshtein)
- CER

$$CER = \frac{S + D + I}{N}$$

Где S - количество замен

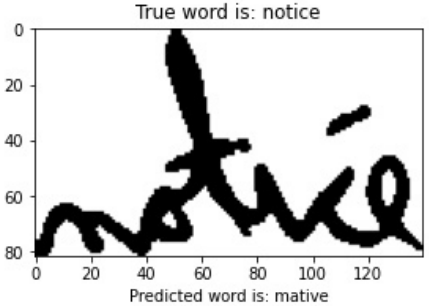
D - количество удалений

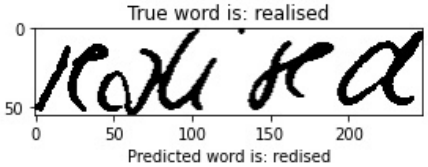
I - Количество вставок

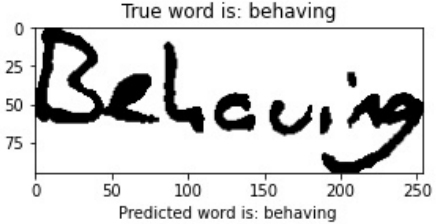
N - количество символов в слове

Пример: “rain” -> “sain” -> “shin” -> “shine” (S = 2, D = 0, I=1, N = 5) => CER = 0.6

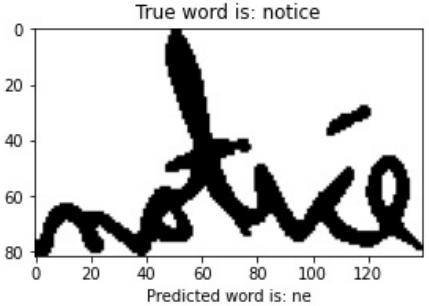
ResNet-BiLSTM-ATTN

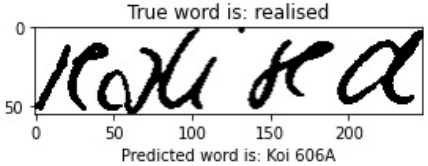


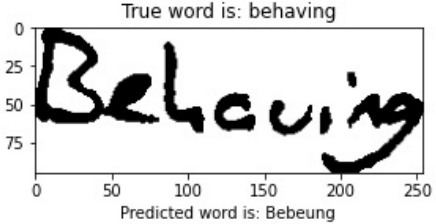




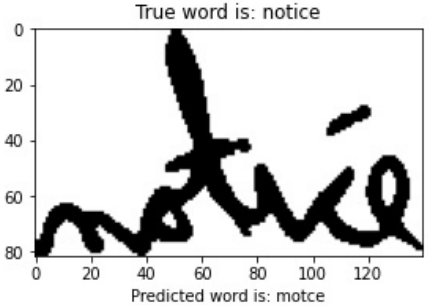
Tesseract

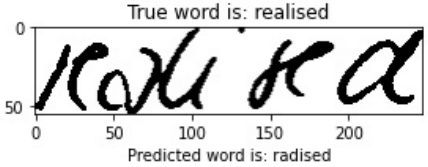


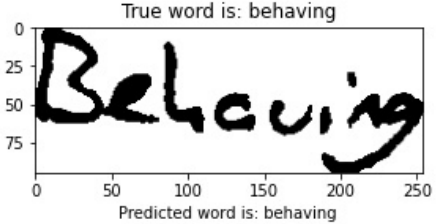




ResNet-BiLSTM-CTC







Использованные источники

1. IAM dataset https://fki.tic.heia-fr.ch/databases/iam-handwriting-database
2. Tesseract engine https://nanonets.com/blog/ocr-with-tesseract/amp/&ved=2ahUKEwjwxuLWrMv0AhVv-SoKHY-NCkMQFnoECCYQAQ&usg=AOvVaw2g68r7GTHRmqECb8HV9fJl
3. Tesseract engine, 2007 - https://static.googleusercontent.com/media/research.google.com/ru//pubs/archive/33418.pdf
4. Tesseract modernization https://github.com/tesseract-ocr/docs/blob/main/das_tutorial2016/6ModernizationEfforts.pdf
5. 3-stage model. Repository https://github.com/clovaai/deep-text-recognition-benchmark
6. 3-stage model. Paper https://arxiv.org/pdf/1904.01906v4.pdf
7. Attention paper https://arxiv.org/pdf/1409.0473.pdf
8. SynthText dataset https://www.robots.ox.ac.uk/~vgg/data/scenetext/
9. MJSynth dataset https://www.robots.ox.ac.uk/~vgg/data/text/