

Пояснительная записка

к дипломной работе на тему:

**"Распознавание текста в документах"**

Автор: Бондарь Андрей

Группа: DLL-18

# ОГЛАВЛЕНИЕ

[ПОСТАНОВКА ЗАДАЧИ](#_heading=h.spc3uk2up34q) **3**

[ДАННЫЕ](#_heading=h.q13f046s84sf) **4**

[МЕТОДИКА РЕАЛИЗАЦИИ](#_heading=h.ntppztvd7nas) **7**

ИТОГИ ОБУЧЕНИЯ

[ВЫВОДЫ](#_heading=h.rbz00x7s2ov4) **11**

# ПОСТАНОВКА ЗАДАЧИ

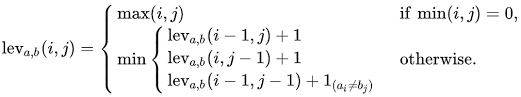
Задача распознавания текста в документах является весьма распространенной задачей. Переход от бумажного и аналогового способа хранения документов к цифровому имеет ряд преимуществ. Во-первых, снижаются затраты на хранение документов в физической форме. Во-вторых, значительно облегчается поиск необходимой информации в цифровой форме. В-третьих, открываются огромные возможности для анализа данных, в том числе с использованием методов ИИ. Для решения задачи были выбраны модели yolo-v5 для детекции необходимых областей и слов, а в качестве распознавания были протестированы две модели классическая CRNN[[1]](#footnote-1) и новая архитектура от апреля 2022 г. SVTR[[2]](#footnote-2).

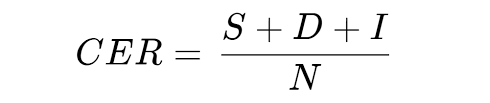
Используемые метрики:

Для детекции - mAP50, mAP95:



Для распознавания - Levenshtein distance, CER:





# ДАННЫЕ

Для дообучения модели первого уровня детекции необходимых областей использовались 63 вручную размеченных изображения. Для разметки использовалась open source библиотека label-studio (рис. 1). В качестве валидационной выборки использовались те же данные.

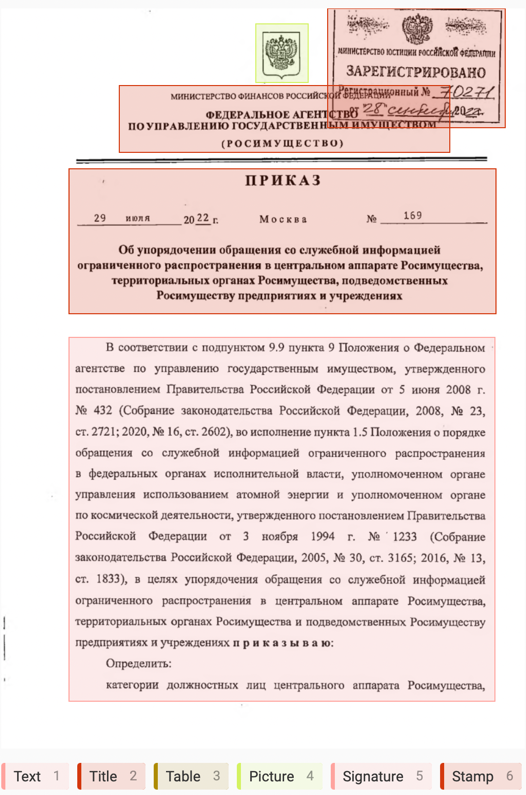


Рис. 1

Для дообучения модели второго уровня детекции использовались 115 размеченных с помощью существующей модели, а затем доразмеченных вручную с помощью label-studio данных (рис. 2). В качестве вадидационной выборки использовались те же данные.

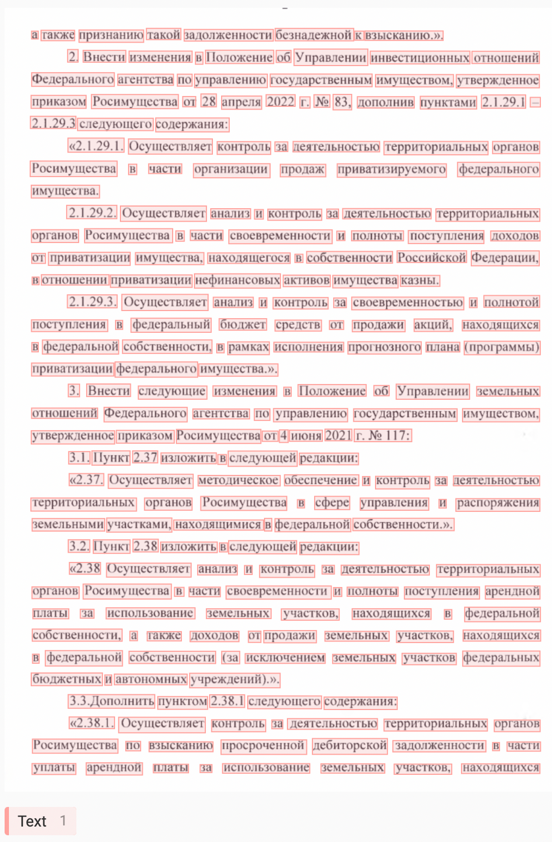


Рис. 2

Модель распознавания тренировалась на 244 тысячах сгенерированных изображениях. Для генерации применялась библиотека “text\_renderer”[[3]](#footnote-3). На рисунке № 3 представлены примеры сгенерированных изображений с текстом. Также применялись функции на python для генерации различных вариаций слов (upper case, lower case, numbers), и дополнительно в “text\_renderer“ применялось 80 различных шрифтов, а также различное расширение изображений.

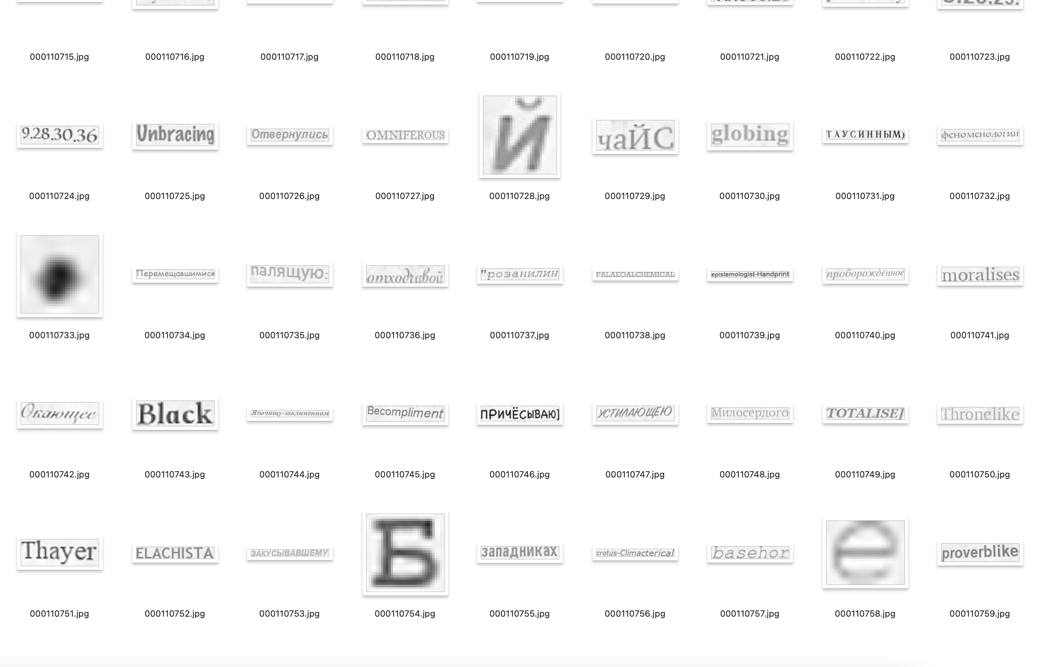


Рис. 3

 В качестве валидационной выборки распознавания текста использовались 1129 реальных случайных вырезанных слов из 63 изображений с помощью модели детекции 1-го уровня, а затем 2-го уровня (рис. 4).



Рис. 4

В качестве тестовой выборки всего пайплайна (det1-> det2-> rec) использовалась одна страница изображения с количеством символов в ground truth – 1644 (рис. 5).

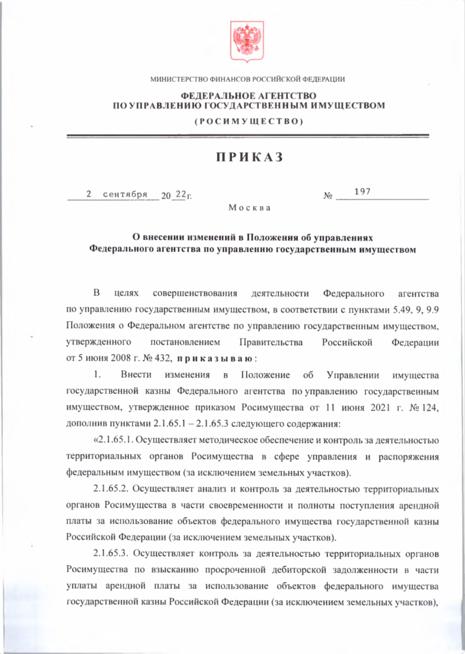


Рис. 5

# Методика реализации

В качестве моделей детекции 1-го и второго уровней использовались модели yolo-v5-large.

Основным в архитектуре YOLOv5 является то, что YOLOv5 относится к архитектуре One-Stage detector - подход, который предсказывает координаты определённого количества bounding box с результатами классификации и вероятности нахождения объекта, и в дальнейшем корректируя их местоположение. В целом такую архитектуру можно представить в следующем виде:

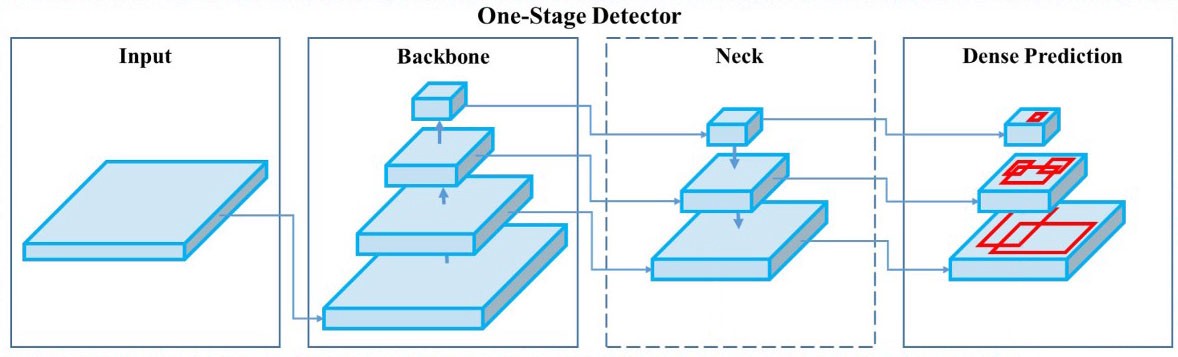


Рис. 5

В качестве моделей распознавания были протестированы два подхода. Первый с применением CRNN и resnet34 в качестве backbone. Архитектура CRNN изображена на рис. 6.

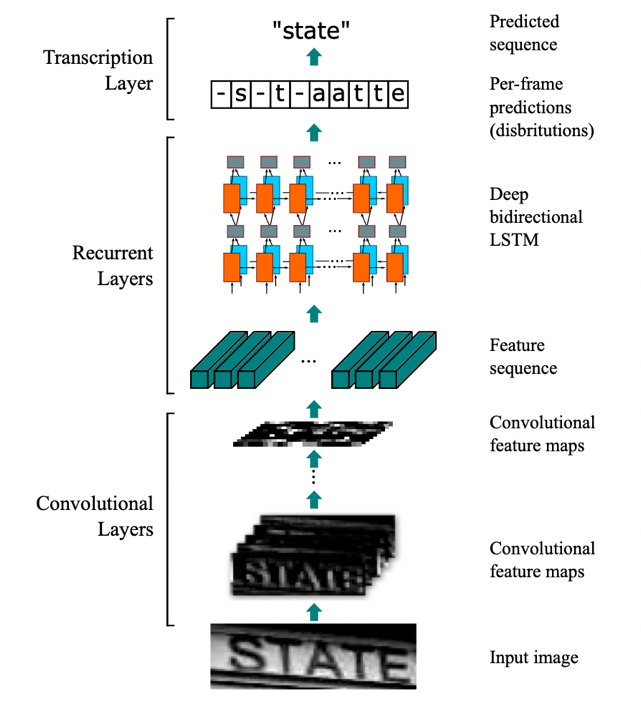


Рис. 6

Во втором подходе использовалась новая архитектура SVTR (рис. 7).

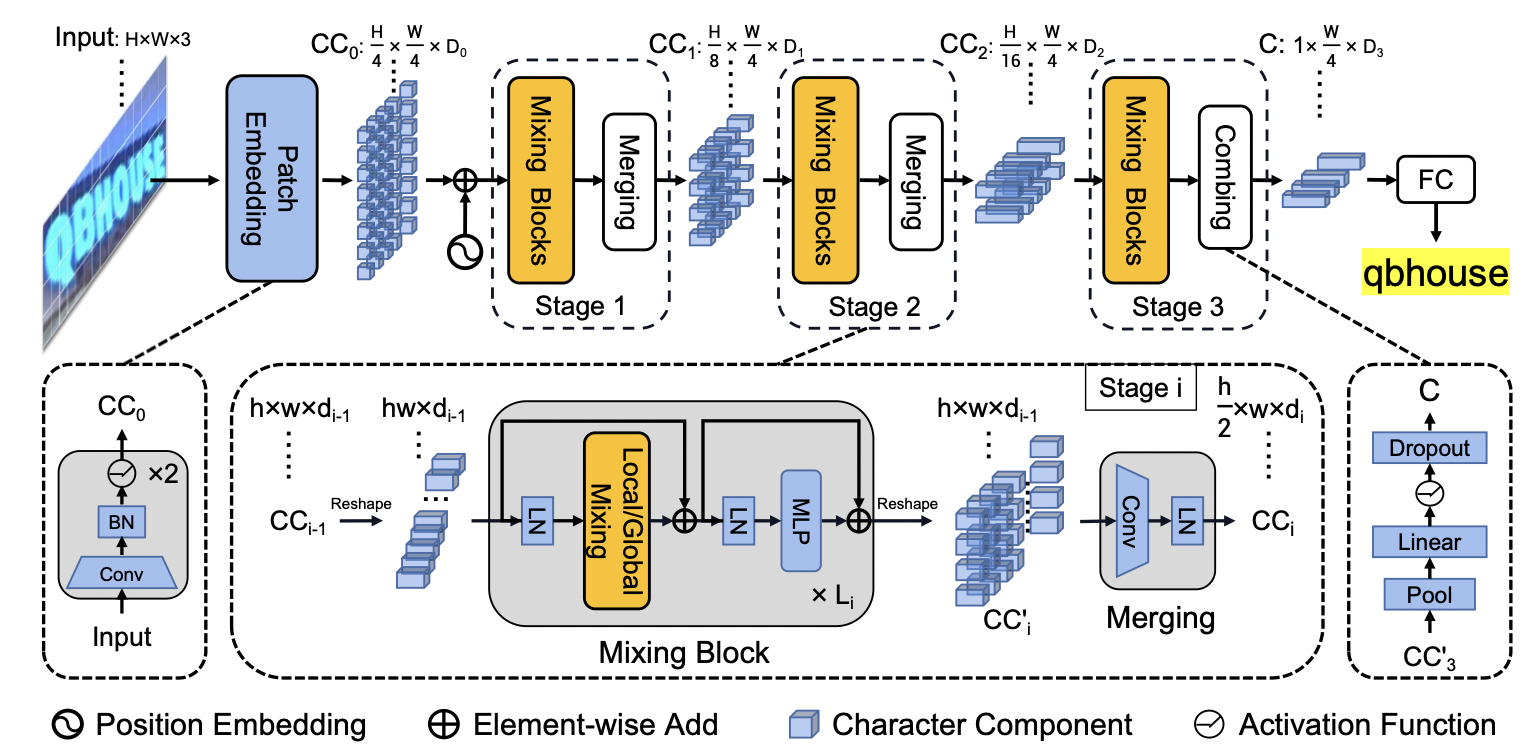


Рис. 7

Основным в данной архитектуре является отказ от использования двоичной архитектуры как в модели CRNN (CNN для извлечения признаков и рекуррентной RNN для предсказания) и переход к цельной, однопроходной модели. В архитектуре SVTR используются Local и Global mixing слои, которые по сути являются multi-head self-attention, и выполняют функцию улавливания глобального и локального контекста на изображении, что в отличии от RNN архитектуры позволяет учитывать контекст изображения более полно.

По итогам обучения моделей были получены следующие метрики на валидационных выборках:

|  |  |  |
| --- | --- | --- |
| Модель | Метрика | Валидационные данные |
| Yolov5-large-1st\_level\_det | mAP50 - 0.991, mAP95 - 0.791 | 63 изображения |
| Yolov5-large-2nd\_level\_det | mAP50 - 0.995, mAP95 - 0.825 | 115 изображений |
| Resnet34 CRNN | Levenstein score - 97.94% | 1129 изображений слов |
| SVTR-Tiny | Levenstein score - 98.5% | 1129 изображений слов |

Таб. 1

# Итоги обучения

В итоге для решения задачи распознавания текста в документах было обучено 4 модели и получено два финальных пайплайна а именно:

Yolov5-large-1st\_level\_det **->** Yolov5-large-2nd\_level\_det **->** Resnet34 CRNN

Yolov5-large-1st\_level\_det **->** Yolov5-large-2nd\_level\_det **->** SVTR-Tiny

В таблице ниже приведены результаты подсчёта метрик для первого и второго пайплайнов:

|  |  |  |
| --- | --- | --- |
| Пайплайн | Levenshtein score в процентах (1644 символов – ground truth) | Скорость работы моделей |
| Resnet34 CRNN | 99.2 % | 3.12 сек/1 it |
| SVTR-Tiny | 99.57 % | 1.98 сек/1 it |

Таб. 2

Ниже представлены замеры скорости работы пайплайнов на 63 тестовых изображениях:

Скорость работы Resnet34 CRNN: Скорость работы SVTR-Tiny:

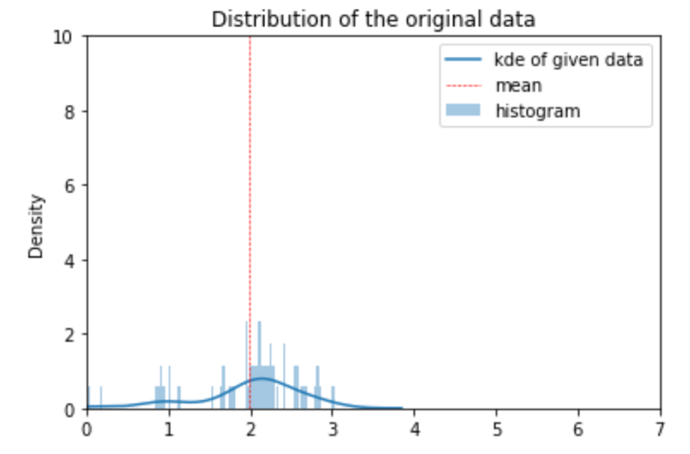
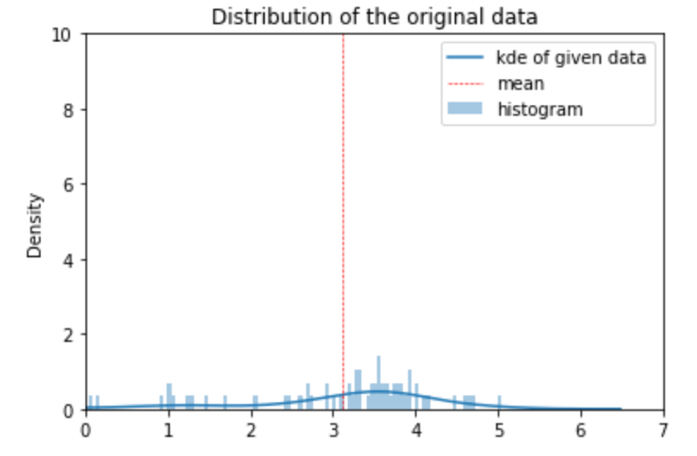


Рис. 8 Рис. 9

# Выводы

Как можно увидеть из таблицы 2 точность модели в которой используется новая архитектура SVTR выше. Из таблицы 1 также видно, что SVTR заметно выигрывает на валидационной выборке. По скорости работы SVTR превосходит CRNN с Resnet34 в качестве backbone. В качестве финального варианта для использования в пайплайне распознавания документов был выбран SVTR-Tiny.

В качестве дальнейших шагов по улучшению моделей детекции могут быть увеличение тренировочных выборок. Возможные шаги по улучшению модели распознавания могут заключатся в использовании цельных строк а не отдельных слов которые передаются модели на распознавание – так возможно контекст который будет поступать модели будет более полным и модель будет меньше ошибаться.

1. https://arxiv.org/pdf/1507.05717.pdf [↑](#footnote-ref-1)
2. https://arxiv.org/pdf/2205.00159.pdf [↑](#footnote-ref-2)
3. https://github.com/Sanster/text\_renderer [↑](#footnote-ref-3)