Отчёт о выполнении тестового задания

1. Запуск проекта

Для запуска проекта необходимо установить требуемые модули, установка модулей производится командной

```
pip install -r requirements.txt
```

Далее запустить скрипт predict.py

```
python predict.py "img path"
```

где img_path – путь до каталога с изображениями в абсолютном виде (D:\img). Данный параметр является обязательным, иначе скрипт оповестит пользователя о неправильности переданного пути и закончит выполнение.

Для каждого изображения скрипт печатает результат предсказаний, а также в конце работы создаётся csv файл, в который сохраняет все предсказания.

2. Обработка данных

Датасет состоит из двух типов изображений: изображения с коротким текстом (от 4 до 6 символов, изображения 280x70) и с длинным (от 1 до 46 символов, 248x81). Так как содержимое и внешний вид данных типов изображений сильно отличается друг от друга, для каждого типа будет использоваться отдельная модель нейронной сети.

Для каждого типа изображений был сформирован словарь символов, которые присутствуют на изображениях, а также длина текста на изображениях

Для коротких текстов, количество символов находилось в диапазоне от 4 до 6, однако один файл с длиной символов 4 и три файла с длиной символов 5 были неверно названы, поэтому данные файлы были вручную переименованы.

С длинными текстами ситуация была хуже, так как в них были изображения с неправильными разметками из-за: отсутствия пробелов между словами, неправильный текста, наличие заглавных букв в разметке. Также в данных присутствовали изображения с длинным текстом, эти изображения были удалены. Суммарно в датасете было удалено 2 изображения, изменено 30 изображений

Далее обрабатывались сами изображения

К изображениям применялся стандартный алгоритм адаптивного порога, также алгоритм Отсу и нормализация значений.

Для изображений с коротким текстом алгоритмы адаптивного порога не сильно улучшали качество изображения и результаты работы модели, поэтому данный тип изображений только нормализовался.

Изображения с длинным текстом являются сильно зашумленными и искажёнными, также текст на изображениях мог быть как белым, так и серым/чёрным, поэтому метод пороговых значений работал плохо. Результаты работы данного метода показаны на рисунке 1

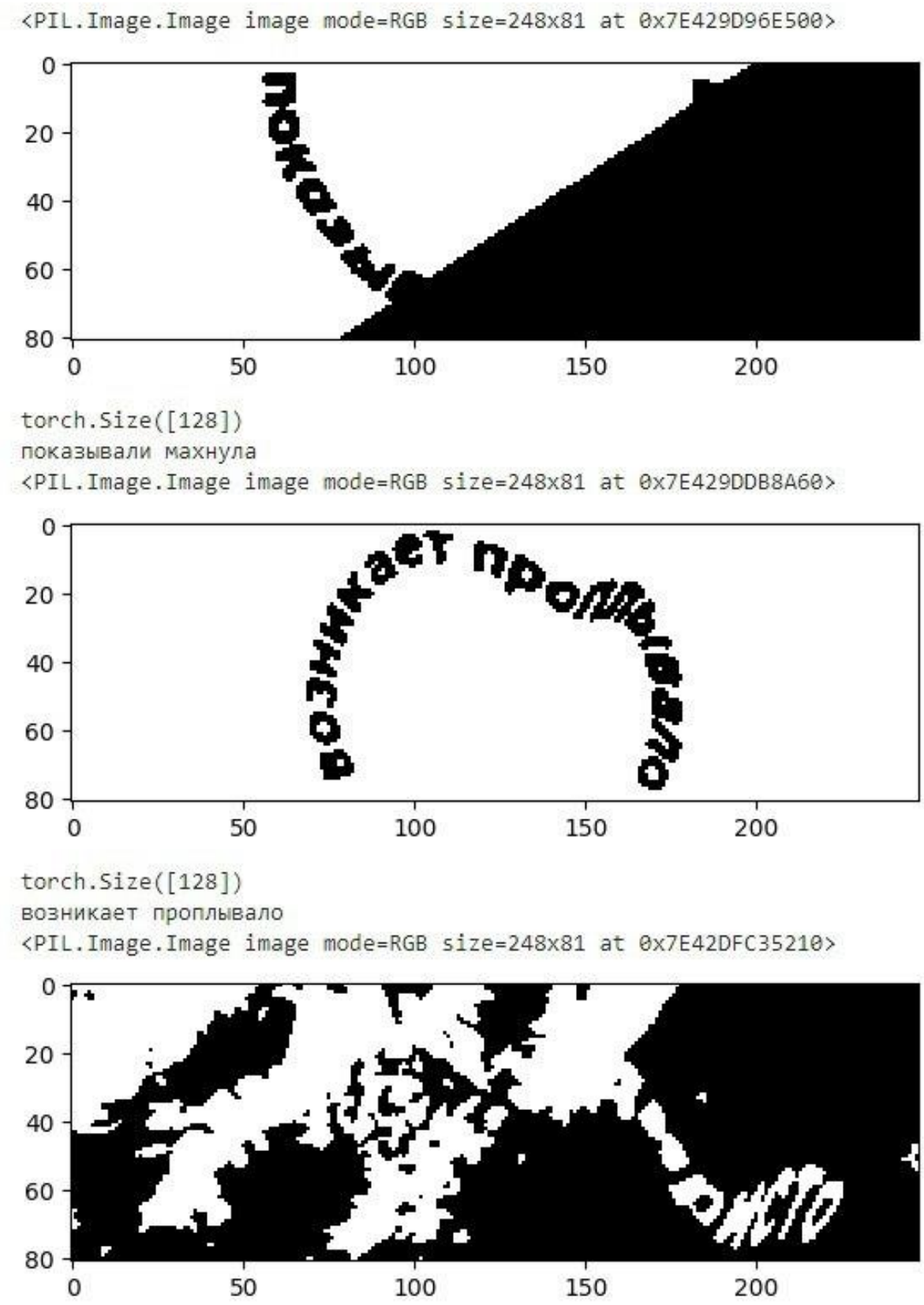

Рисунок 1 – Применение алгоритма Отсу

Данный алгоритм хорошо удаляет посторонний шум, однако из-за наличия большого и разнообразного шума, а также различных цветов текста, алгоритм Отсу перекрывает текст на многих изображениях, поэтому был использован обычный алгоритм пороговых значений.

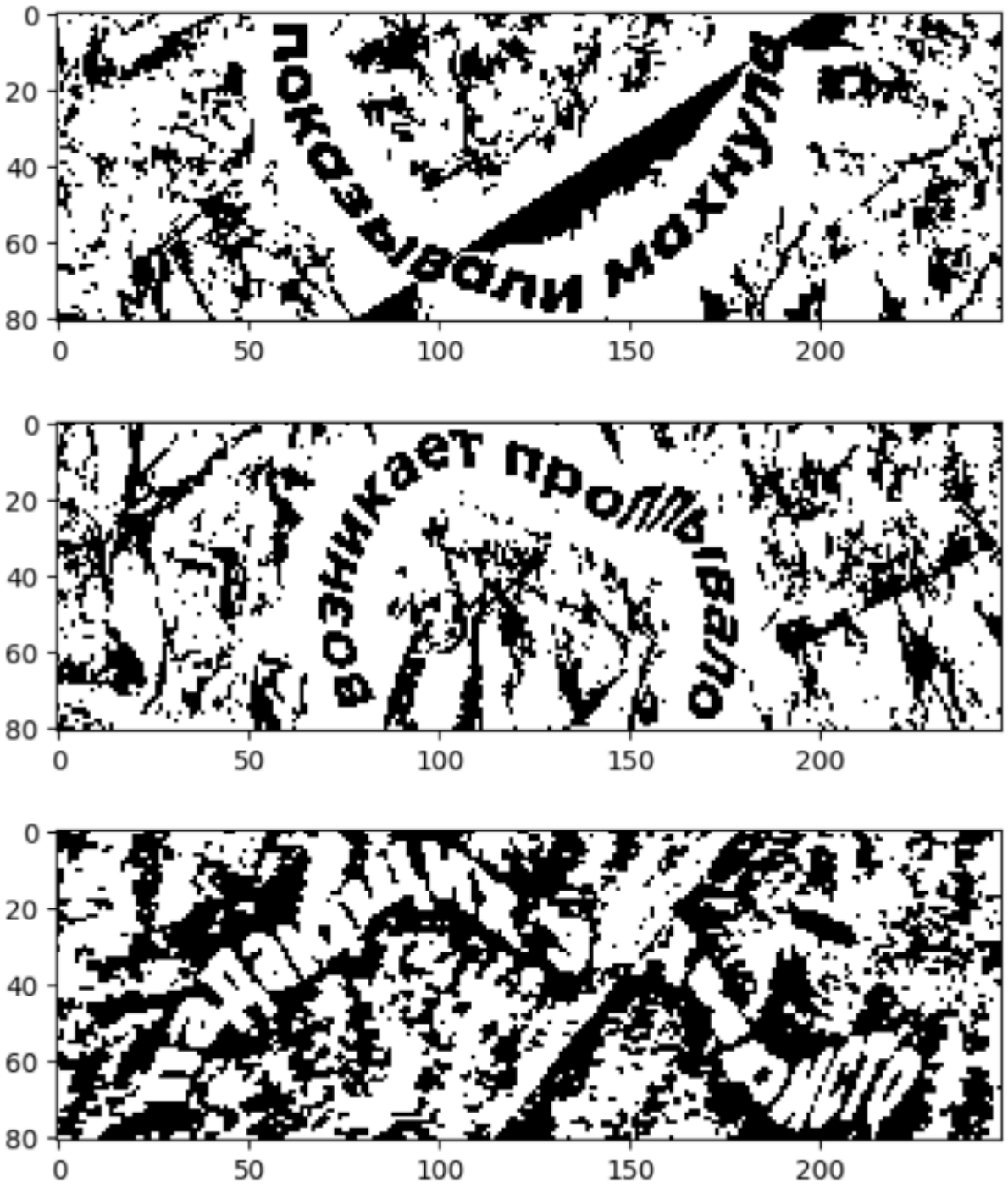

Рисунок 2 – Стандартный алгоритм адаптивного порогового значения

Данный алгоритм также перекрывает текст на изображениях, однако делает он это редко, поэтому ценой нескольких десятков изображений была получена менее зашумлённая, бинаризованная картинка.

3. Модели нейронных сетей

Изначально была реализована CRNN, состоящая из 3 блоков:

- Блок свёрточной сети, которая выделяла основные признаки из изображений
- Рекуррентный слой(2 слоя двунаправленных LSTM), который формировал последовательность кодов символов с изображения
- Декодеровщик, который преобразовывал коды в текст

В качестве функции ошибки использовалась CTC ошибка (чем меньше ошибка, тем лучше).

Архитектура данной модели показана на рисунке 3

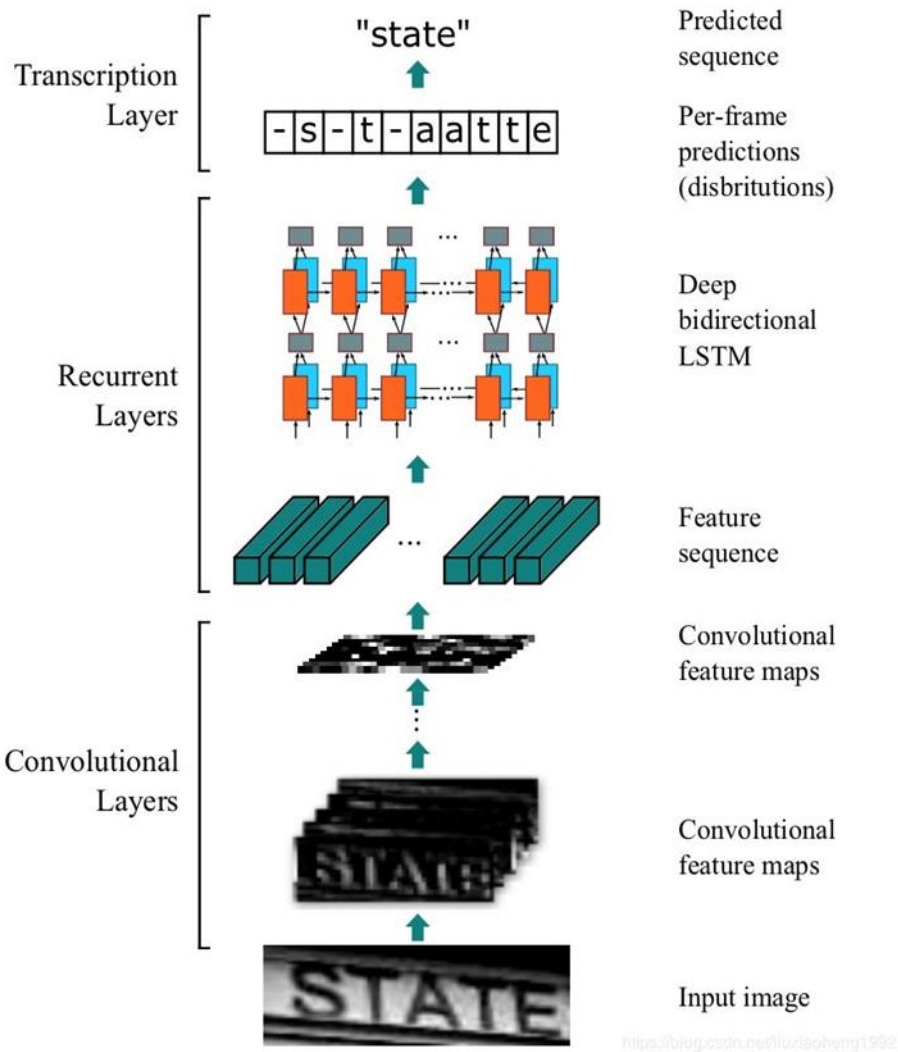

Рисунок 3 – Архитектура CRNN

Данная модель была отдельно обучена на длинных и коротких текстах. В результате обучения на коротких текстах модель показала лучшую ошибку в 0.48 единиц. График обучения показан на рисунке 4.

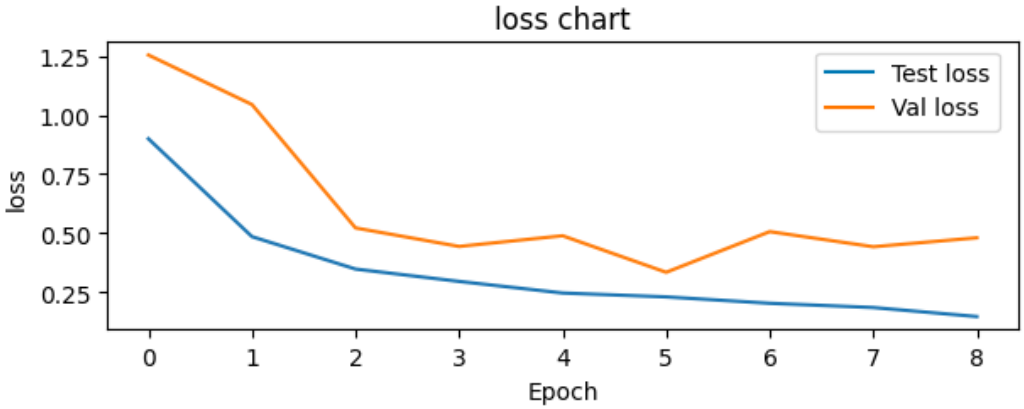

Рисунок 4 – Обучение CRNN на коротких текстах

Далее для оценки качества генерации был высчитана CER метрика, которая показывает процентное количество неверно распознанных символов. Величина CER для коротких текстов равнялась 0.02. Модель ошибается в 2% символах. Примеры распознавания показаны в таблице 1

Таблица 1 – Результаты распознавания

Реальный текст	Выход модели	CER (доли)
yutp9h	yutp9h	0.0
kpnpxd	kpnpxu	0.1667
rfgxrt	rfgxrt	0.0
saxuzy	saxuzy	0.0
djelwk	djelwk	0.0
cjp4wy	cjp4wy	0.0

Так как данная модель показала хороший результат и в качестве распознавания, и в скорости распознавания (1 мс на изображение), то последующее тестирование моделей производилось только на данных с длинным текстом.

Датасет с длинным текстом содержит более сложные данные, поэтому для него было использовано три модели: CRNN, TrOCR (модель на базе визуального трансформера) и CRNN + STN. Архитектуры TrOCR и CRNN + STN показаны на рисунках 5 и 6.

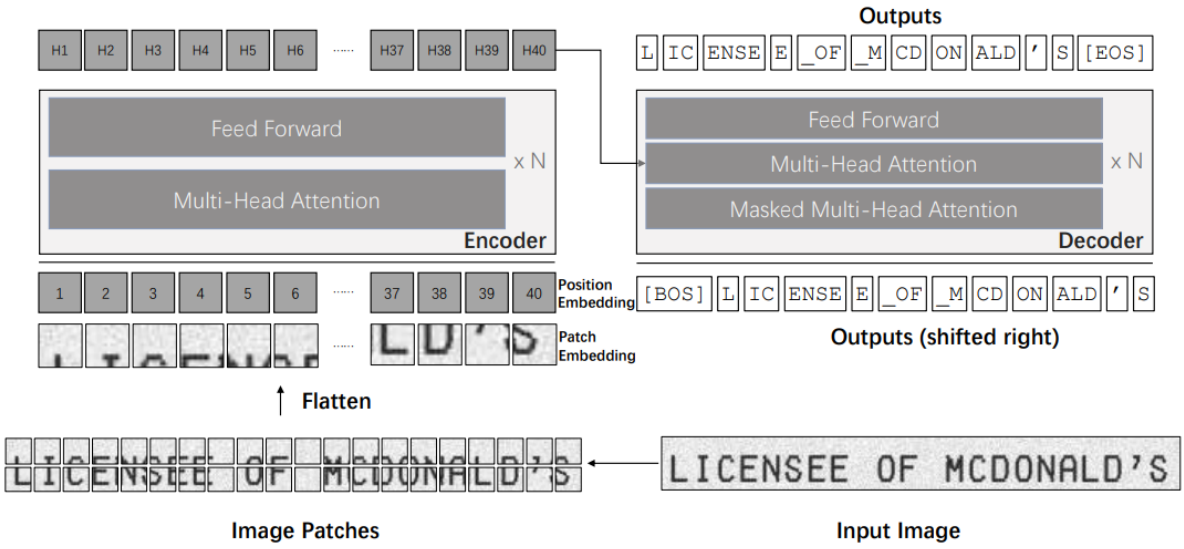

Рисунок 5 – Архитектура TrOCR

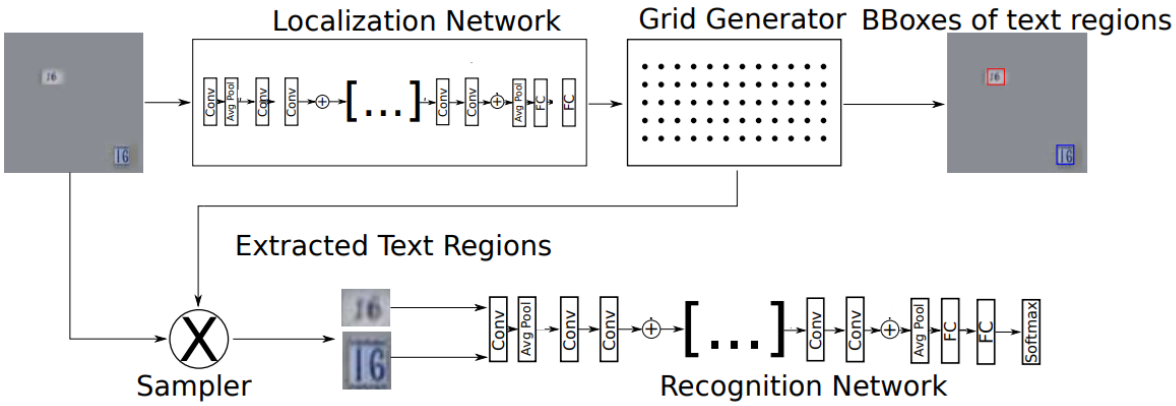

Рисунок 6 – CRNN + STN

Первой обучалась CRNN модель. Обучение длилось 100 эпох. График обучения показан на рисунке 7

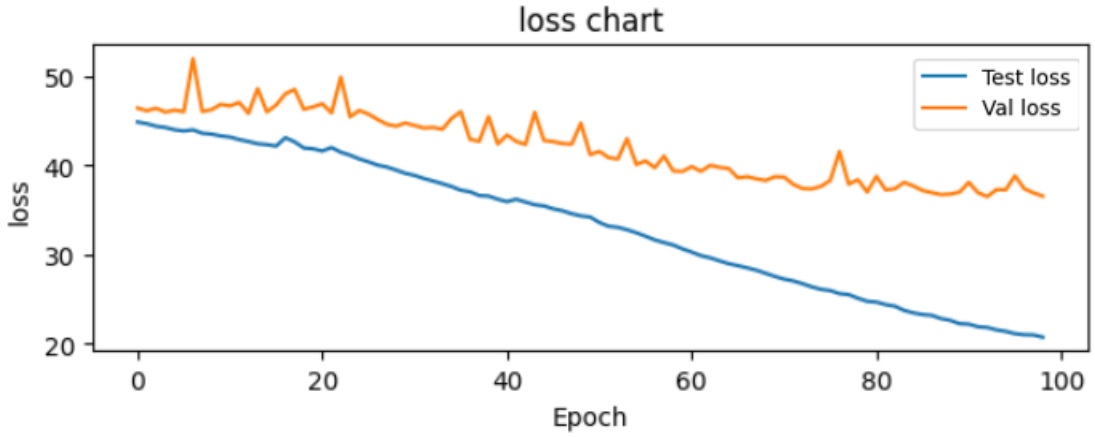

Рисунок 7 – График обучения модели CRNN

Лучшее значение ошибки на валидационной выборке равнялось 36.7, после сотой эпохи модель стала переобучаться и лучшего значения не было получено. Послее обучения модель выдаёт среднее значение CER в 54%, что очень много. Результаты предсказаний показаны в таблице 2

Текст на изображениях	Предсказание модели	CER (доли)
сложив тарахтит	сложица раратkит	0.33
крышах фигурными	рава раравены	0.75
небольшой поданы	ерель пооаних	0.56
летучих прояснить	лелдих просниль	0.29

Модель показала посредственные результаты, поэтому были проведены эксперименты с другими моделями.

Скорее всего данная модель не может обучиться на данных, так как текст на изображения загнут, а также даже после очисти изображения от шума, количество шумных пикселей достаточно большое. Вторую проблему можно решить с помощью дополнительной модели, которая будет преобразовывать изображения в чёрно-белый формат. Первую проблему можно попытаться решить с помощью архитектур нейронных сетей, например CRNN + STN. Поэтому после обучения первой модели, были обучены ещё две.

Обучение данных моделей ничего не дало, значение ошибки за 40 эпох изменилось на 5 единиц, с 53 до 48 для CRNN + STN и с 50 до 45 для TrOCR, после этого улучшения качества распознавания не было и модели выдавали невнятные предсказания.

4. Выводы и улучшения

Получилось реализовать работоспособную модель для распознавания изображений с короткими текстами, однако полноценно решить задачу распознавания изображений с длинными текстами не получилось

Улучшения

1) Так как капчи могут иметь различные размеры, необходимо будет добавить предобработку изображений по изменению их размера.
2) На изображения с длинными текстами присутствует большое количество шумов, также текст на изображениях попадается как белым цветом, так и серым или чёрным, поэтому необходимо реализовать более продвинутый механизм бинаризации, чем используется в данном решении. Один из вариантов – реализовать нейронную сеть для бинаризации
3) Так как для различных типов капч в данном решении используются различные модели, необходимо будет реализовать классификатор, который будет определять тип каптчи
4) Основной сложностью при распознавании длинных текстов является наличие искажения форм текста. Модель на основе STN предназначена для выравнивания таких текстов, однако

реализованная версия недостаточно адаптирована под такие сильные искажения, поэтому для улучшения качества распознавания будет увеличено количество контрольных точек, по которым происходят преобразования, также свёрточные блоки будут заменены на Res-блоки с остаточным соединением, чтобы модель лучше сходилась и быстрее обучалась.