Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра теоретической и прикладной информатики

ОТЧЕТ ПО ПРАКТИКЕ

Производственная практика: технологическая (проектно-технологическая) практика
(наименование практики в соответствии с учебным планом)

Направление подготовки: 01.03.02 Прикладная математика и информатика

Выполнил:

Студент Иванов В.В.
(Ф.И.О.)

Группа ПМ-91

Факультет ПМИ

_____
подпись

«14» апреля 2023 г.

Проверил:

Руководитель от НГТУ Попов А.А.
(Ф.И.О.)

Балл: _____, ECTS_____,

Оценка _____
«отлично», «хорошо», «удовлетворительно», «неуд.»

_____
подпись

«15» апреля 2023 г.

Новосибирск 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ В АРХИТЕКТУРУ ТРАНСФОРМЕРА

Трансформер - это архитектура нейронной сети, разработанная для обработки последовательностей, таких как тексты или звук. Она была представлена в статье "Attention Is All You Need", опубликованной в 2017 году исследователями из Google Brain.

Трансформеры быстро стали одной из самых популярных архитектур для обработки естественного языка. Они обеспечивают высокую точность и эффективность, что делает их подходящими для решения широкого круга задач, связанных с обработкой естественного языка. Самое большое преимущество трансформеров по сравнению с рекуррентными нейронными сетями заключается в их высокой эффективности в условиях параллелизации.

Основной принцип работы трансформеров - это механизм внимания (attention mechanism), который позволяет нейросети сфокусироваться на наиболее важных частях последовательности. В традиционных рекуррентных нейронных сетях (RNN) информация передается от одного шага к следующему, что может приводить к проблемам с долгосрочными зависимостями. В трансформерах же информация передается параллельно, что позволяет сети более эффективно улавливать длинные зависимости.

Механизм внимания работает следующим образом: для каждого элемента последовательности сначала вычисляется вектор запроса (query), который используется для нахождения наиболее релевантных элементов последовательности. Затем вычисляются векторы ключей (keys) и значения (values), которые описывают каждый элемент последовательности. После этого для каждого элемента последовательности вычисляется весовой коэффициент, отражающий степень важности этого элемента для данного запроса. Взвешенная сумма всех значений дает выход механизма внимания.

Главным элементом трансформера является блок трансформации (transformer block), который содержит несколько слоев механизма внимания и полносвязных слоев (feedforward layers). Каждый блок трансформации

обрабатывает входную последовательность и передает ее на следующий блок, постепенно улучшая качество предсказаний. Полносвязный слой в блоке трансформации на вход принимает выход механизма внимания и применяет к нему набор линейных преобразований и нелинейных функций активации, чтобы получить финальный выход блока.

Трансформеры используют технику под названием “многозаголовочное внимание” (multi-head attention), которая позволяет нейросети использовать несколько параллельных механизмов внимания, каждый из которых имеет свои векторы запросов, ключей и значений. Это позволяет нейросети более эффективно моделировать зависимости во входных данных. Также в этой архитектуре используют дополнительные компоненты, такие как позиционные кодировки (positional encodings), которые добавляются к входным данным, чтобы сохранить информацию о порядке элементов последовательности, и нормализацию слоев (layer normalization), которая помогает ускорить обучение и улучшить стабильность обучения.

Кроме того, трансформеры обучаются с помощью метода самообучения (self-supervised learning), что означает, что модель обучается на большом наборе данных, но без явной разметки. Вместо этого модель сама создает задачи для обучения, такие как предсказание следующего слова в тексте, и использует их для обучения. Этот подход позволяет использовать гораздо больше данных, чем если бы необходима явная разметка.

Трансформеры состоят из двух основных компонентов: трансформера-кодировщика и трансформера-декодировщика. Трансформер-кодировщик обрабатывает входную последовательность и преобразует ее во внутреннее представление, которое затем передается в трансформер-декодировщик для генерации выходной последовательности.

Трансформер-кодировщик состоит из нескольких слоев, каждый из которых содержит два основных блока: многозаголовочное внимание и полносвязный слой, называемый позиционно-возможностным слоем (Position-wise Feedforward Layer). Многозаголовочное внимание позволяет

модели обрабатывать информацию из разных частей входной последовательности и создавать связи между ними. Позиционно-возможностной слой применяет линейное преобразование к каждой позиции входной последовательности независимо от других позиций.

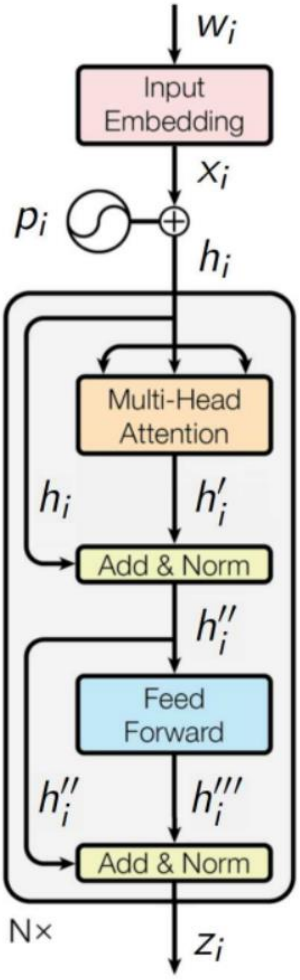

Рис. 1 – Архитектура трансформера-кодировщика

Трансформер-декодировщик также состоит из нескольких слоев, каждый из которых содержит те же два блока: многозаголовочное внимание и позиционно-возможностный слой. Кроме того, трансформер-декодировщик также содержит третий блок, называемый маскированным многозаголовочным вниманием (Masked Multi-Head Attention), который позволяет модели генерировать выходную последовательность постепенно, учитывая только те части входной последовательности, которые были уже

обработаны. Это необходимо для задач, где выходная последовательность генерируется последовательно. [1]

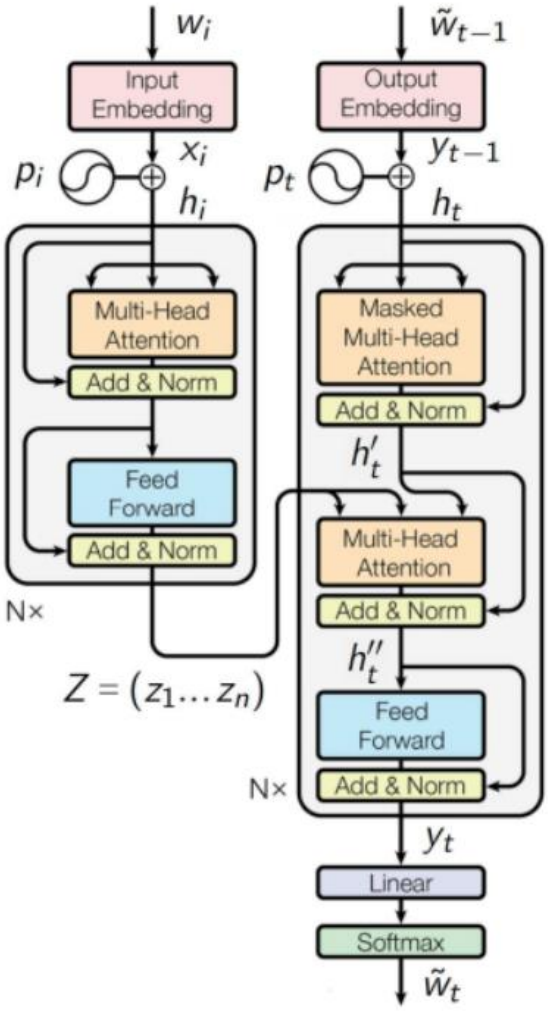

Рис. 2 – Архитектура трансформера-декодировщика

ТРАНСФОРМЕРЫ В ЗАДАЧЕ OCR

В задаче оптического распознавания символов, где необходимо распознать текст на изображении, архитектура трансформера может быть использована для обработки изображения, представленного в виде последовательности пикселей. В этом случае механизм внимания может быть использован для сосредоточения на определенных частях изображения, которые могут содержать символы или другую важную информацию. Каждый пиксель рассматривается как отдельный элемент последовательности, а модель обрабатывает их последовательно, чтобы извлечь признаки, которые могут быть использованы для распознавания символов.

Существует множество моделей трансформеров, используемых для обработки изображений. Рассмотрим те, которые были разработаны специально для решения задачи обнаружения текста на изображениях.

TROCR

TrOCR - это модель распознавания, не способная детектировать текст. Она состоит из двух частей:

1. ViT (Vision Transformer) - в качестве кодировщика для работы с изображением
2. RoBERTa (Robustly Optimized BERT Pretraining Approach) - в качестве декодировщика для работы с текстом

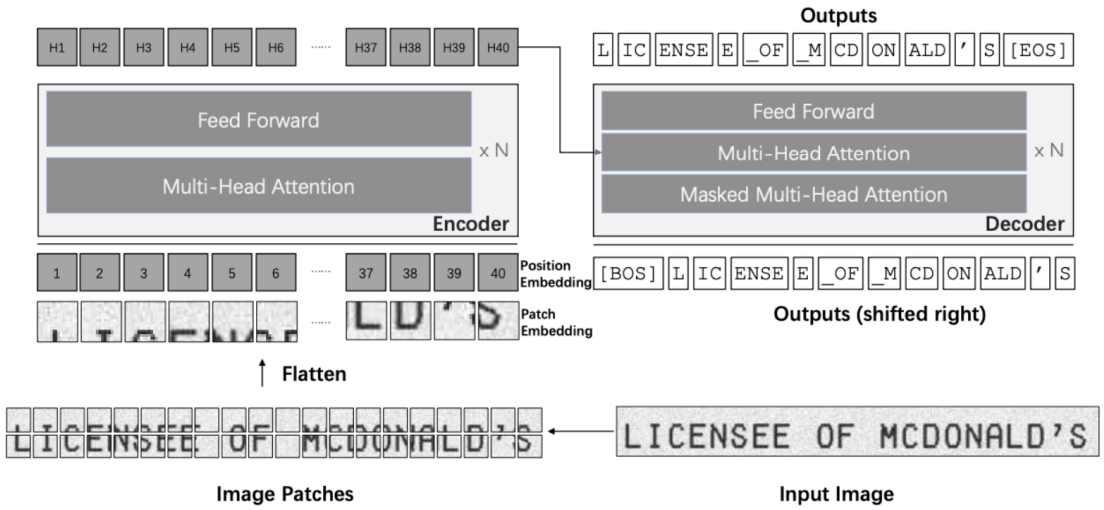

Рис. 3 – Архитектура TrOCR

Основная идея TrOCR заключается в использовании предварительно обученных моделей трансформеров для обработки изображений и генерации текста на уровне слов. Это позволяет создать эффективную и простую модель, которая может быть обучена на больших объемах синтетических данных и дообучена на реальных данных, размеченных людьми.

Большая часть обучения модели происходила в режиме обучения без учителя, либо на синтетически сгенерированных изображениях строк. В конце производилось дообучение модели на одну из двух подзадач: распознавание печатного или рукописного текста. [2]

Модель была представлена 6 сентября 2022 года в статье “TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models”. В

ней авторы приводят результаты тестирования на датасете IAM с использованием метрики character error rate, где сравниваются различные модели прошлых годов:

Model	Architecture	Training Data	External LM	CER
$\text{TrOCR}_{\text{BASE}}$	Transformer	Synthetic + IAM	No	3.42
$\text{TrOCR}_{\text{LARGE}}$	Transformer	Synthetic + IAM	No	2.89
(Bluche and Messina, 2017)	GCRNN / CTC	Synthetic + IAM	Yes	3.2
(Michael et al., 2019)	LSTM/LSTM w/Attn	IAM	No	4.87
(Wang et al., 2020)	FCN / GRU	IAM	No	6.4
(Kang et al., 2020)	Transformer w/ CNN	Synthetic + IAM	No	4.67
(Diaz et al., 2021)	S-Attn / CTC	Internal + IAM	No	3.53
(Diaz et al., 2021)	S-Attn / CTC	Internal + IAM	Yes	2.75
(Diaz et al., 2021)	Transformer w/ CNN	Internal + IAM	No	2.96

Рис. 4 – Таблица результатов тестирования различных моделей на датасете IAM

Использование архитектуры ViT в качестве кодировщика обусловлено её очевидными преимуществами перед традиционными архитектурами сверточных нейронных сетей, таких как ResNet или Inception. Она обучается быстрее и достигает лучших результатов на некоторых задачах компьютерного зрения. Основная идея ViT заключается в том, что она представляет изображение в виде плоского вектора, используя механизм самого трансформера для анализа и классификации изображения. Это достигается путем разделения изображения на патчи фиксированного размера, которые затем преобразуются в последовательности и могут быть обработаны трансформерами. [3]

RoBERTa является развитием модели BERT (Bidirectional Encoder Representations from Transformers), которая была выпущена Google в 2018 году. Одной из основных проблем модели BERT была её склонность к переобучению на небольших наборах данных и недостаточной предобработке данных. В отличие от BERT, RoBERTa была предварительно обучена на большем количестве данных и более глубоко предобработана. Модель включает 12 слоев трансформеров и 125 миллионов параметров.

Для обучения RoBERTa было использовано более 160 гигабайт текстовых данных, собранных из Википедии, Common Crawl и других источников. Тексты были предварительно обработаны с помощью различных методов, таких как удаление шума, приведение к нижнему регистру, исправление опечаток, удаление стоп-слов и т.д. После этого текст был разбит на предложения и токенизирован. [4]

SVTR

В статье “SVTR: Scene Text Recognition with a Single Visual Model” 30 апреля 2022 года авторы данной модели предложили архитектуру для распознавания текста на изображениях, которая использует один "зрительный" модуль. Основная идея заключается в обработке признаков разных уровней, таких как локальные признаки, относящиеся к отдельным частям символов, и глобальные признаки всего изображения. Входное изображение сначала разбивается на компоненты, соответствующие частям изображения, и модель извлекает важную информацию, используя механизм self-attention между компонентами. Затем формируется многогранное представление текста на изображении, уменьшается размерность и объединяются признаки после блоков self-attention. В итоге модель выдает последовательность признаков, в которой уже закодирован текст.

Перед подачей изображения на вход нейронной сети используется блок STN (Spatial Transformer Network), который выполняет выпрямление текста на исходном изображении. Затем изображение с выпрямленным текстом разбивается на патчи и каждый патч представляется в виде вектора. Для передачи информации о позиции каждого патча на изображении к выходу Patch Embedding добавляется position embedding. Полученный набор компонент представляет собой тензор определенного размера.

Проблемы, возникающие при распознавании текста, включают неправильную классификацию похожих символов и шум в выделенных признаках. Для качественного распознавания текста авторы статьи предложили, как уже было сказано, выделять локальные и глобальные признаки. Локальные признаки закодированы морфологическими признаками и корреляцией между компонентами определенной части изображения, а глобальные признаки определяют зависимости между всеми компонентами. Для извлечения признаков были разработаны global mixing и local mixing слои. Global mixing представляет собой multi-head self-attention

слой, a local mixing вычисляет взаимосвязь только с соседними компонентами. Тензор, полученный от Patch Embedding, проходит 3 стадии обработки, каждая из которых состоит из mixing blocks, layer normalization и многослойный перцептрон. После обработки последним многослойным перцептроном результатом является тензор, содержащий выходные векторы, соответствующие каждому символу на изображении.

Для уменьшения вычислительной нагрузки и изучения изображения в разном масштабе, полученный тензор проходит блок merging, который уменьшает высоту в 2 раза и увеличивает количество каналов. После двух блоков mixing и merging, на заключительной стадии merging блок заменяется на combing, который формирует тензор с размером (1, W/4, D3). Затем последует полносвязный слой, который предсказывает последовательность размером W/4, состоящую из N компонентов, где N - количество символов для предсказания. [5]

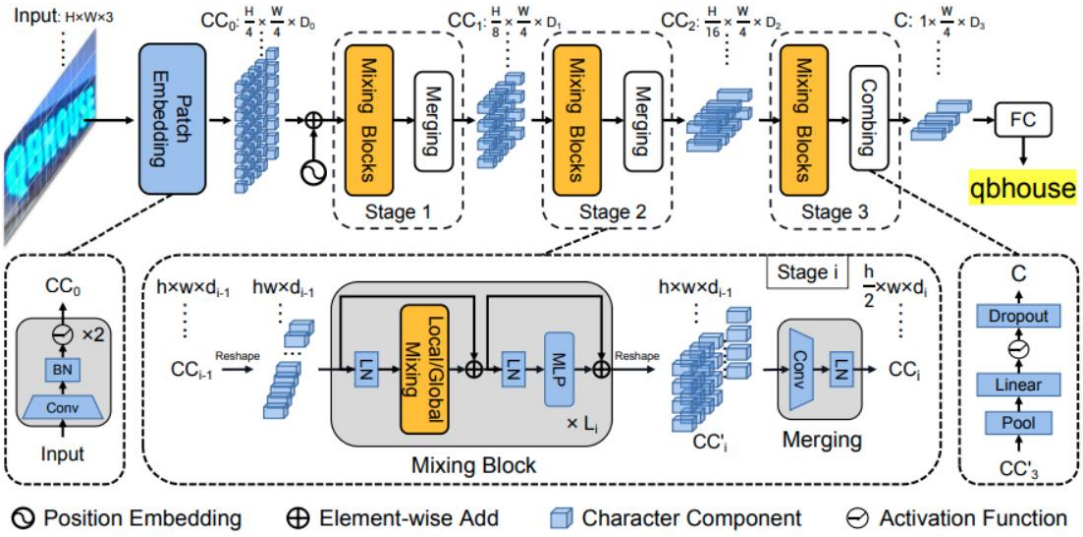

Рис. 5 – Архитектура SVTR

LAYOUTLM

LayoutLM - это модель, которая была создана для распознавания и классификации различных компонентов макетов документов. Впервые была представлена командой исследователей из компании Microsoft на конференции, посвященной обработке естественного языка и компьютерной лингвистике в 2019 году. Основана на архитектуре трансформера и является многоуровневой сетью, которая обрабатывает текст и изображения на разных уровнях. Кроме того, модель использует специальные механизмы внимания и механизмы мульти-задачного обучения для обработки документов различных типов и форматов. Основные модули включают в себя:

1. Модуль трансформера для обработки текста: этот модуль используется для кодирования текста, извлечения признаков и классификации компонентов документа, таких как заголовки, абзацы и списки. Он может работать с различными языками и обрабатывать тексты разной длины.
2. Модуль сверточной нейронной сети для обработки изображений: этот модуль используется для извлечения признаков из изображений, таких как таблицы, графики и изображения. Он преобразует изображение в последовательность признаков и передает ее на вход модулю трансформера.
3. Модуль обучения с подкреплением: этот модуль используется для улучшения точности определения компонентов документа. Он обучает модель на основе награды за правильную классификацию компонентов, что позволяет улучшать предсказания в ходе обучения.

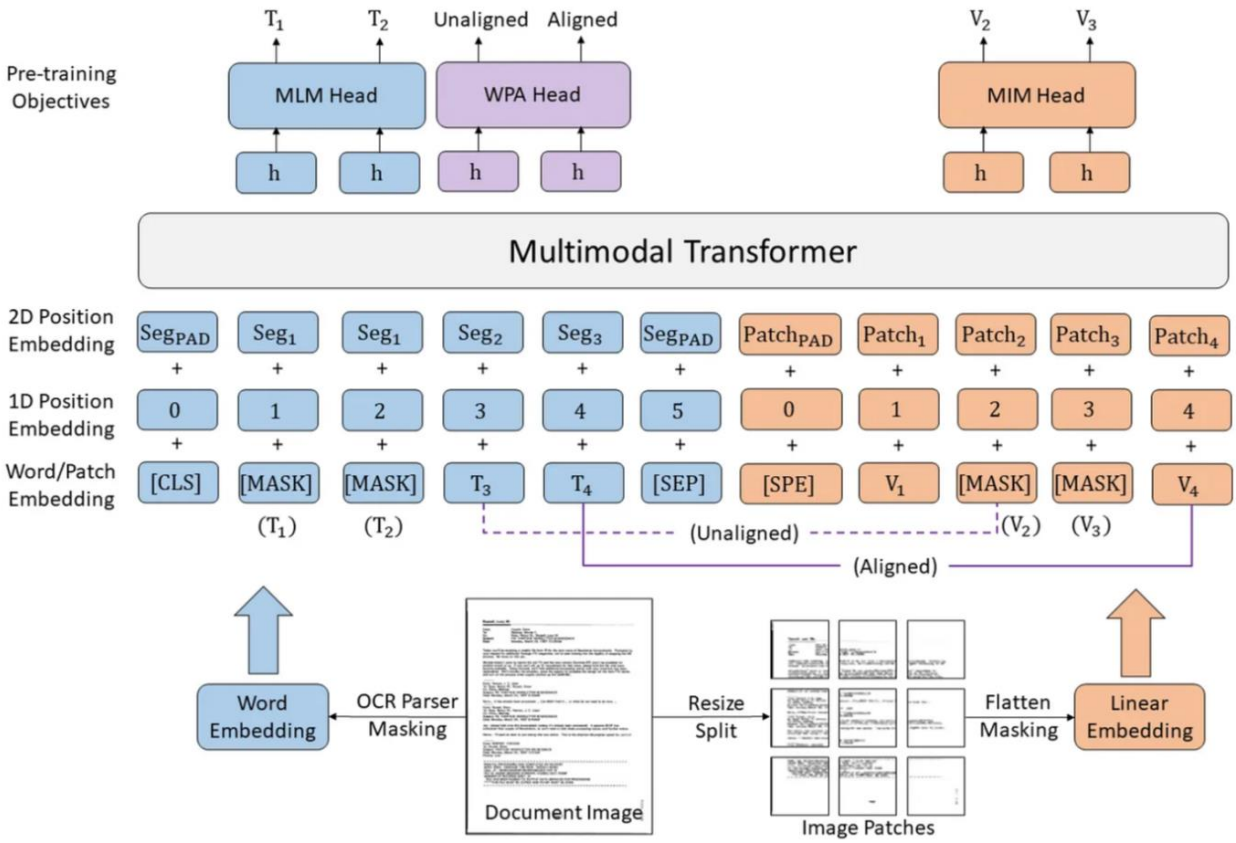

Рис. 6 – Архитектура LayoutLM

Одной из главных особенностей модели является то, что она способна анализировать макеты документов и изображений с высокой точностью в режиме реального времени даже в том случае, если изображения имеют сложную иерархическую структуру и содержат разные типы объектов, такие как текст, таблицы, изображения и диаграммы. [6]

ЗАКЛЮЧЕНИЕ

В ходе выполнения практики было приобретено понимание основных составляющих архитектуры трансформера, а также изучены три актуальных модели, которые были созданы специально для решения задачи оптического распознавания символов.

Кроме того, были выявлены преимущества и недостатки данной архитектуры. Высокая точность достигается за счет того, что трансформеры способны учитывать контекст и связи между символами в словах и предложениях, обрабатывать последовательности произвольной длины, не требуя фиксированного размера входных данных. Кроме того, благодаря своему параллелизму, они обучаются на больших объемах данных, что позволяет повысить эффективность и точность.

К недостаткам можно отнести высокую вычислительную сложность, особенно при обработке больших последовательностей, требовательность к объему памяти и чувствительность к выбору гиперпараметров. Однако, когда речь идёт о применении трансформеров в OCR, первые два недостатка не проявляют себя существенно.

Трансформер является одной из самых эффективных архитектур в задаче обработки естественного языка и может быть перенесен на некоторые проблемы компьютерного зрения. Поскольку OCR совмещает в себе две указанные области, в совокупности это делает трансформер одной из самых лучших OCR-архитектур.

СПИСОК ИСТОЧНИКОВ

1. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention Is All You Need [Электронный ресурс]. URL: https://arxiv.org/abs/1706.03762 (дата обращения 08.04.23)
2. Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, Furu Wei, TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models [Электронный ресурс]. URL: https://arxiv.org/abs/2109.10282 (дата обращения 09.04.23)
3. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale [Электронный ресурс]. URL: https://arxiv.org/abs/2010.11929v2 (дата обращения 09.04.23)
4. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach [Электронный ресурс]. URL: https://arxiv.org/abs/1907.11692 (дата обращения 09.04.23)
5. Yongkun Du, Zhineng Chen, Caiyan Jia, Xiaoting Yin, Tianlun Zheng, Chenxia Li, Yuning Du, Yu-Gang Jiang, SVTR: Scene Text Recognition with a Single Visual Model [Электронный ресурс]. URL: https://arxiv.org/abs/2205.00159 (дата обращения 10.04.23)
6. Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, LayoutLM: Pre-training of Text and Layout for Document Image Understanding [Электронный ресурс]. URL: https://arxiv.org/abs/1912.13318 (дата обращения 11.04.23)