Ministry of Science and Higher Education of the Republic of Kazakhstan

SDU University

Mukhtar Bimurat

Book text recognition in Kazakh Language

THESIS

Presented in Partial Fulfilment for the

Degree of Master of Technical Science in Computer Science

(degree code: 7M06102)

Department of Computer Science

Faculty of Engineering and Natural Sciences

Supervisor: **Ardak Shalkarbayuli**

Kaskelen, June 2024

SDU University
Faculty of Engineering and Natural Sciences
Department of Computer Science

Dean of Faculty of Engineering and Natural Sciences

Assistant Professor, PhD Akhmedov Ramis

_____

« 04 » June 2024

Topic of the thesis:
Book text recognition in Kazakh Language

Thesis submitted as part of the requirements for the award of the MSc in
"7M06102 - Computer Science", SDU University

Head of Department Zhanar Mukash

Academic Supervisor Ardak Shalkarbayuli

Master student Mukhtar Bimurat

Kaskelen, 2024

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Mukhtar Bimurat

June 2024

Acknowledgements

I would like to thank my supervisor Ardak Shalkarbayuli for introducing me to NLP research area and for his advice, guidance, and patience during the preparation of this thesis.

Dedication

This thesis is dedicated to:

My parents, supervisor, and many others for their support, help, sense of humor, and useful comments for improving this project.

Abstract

The digitization of Kazakh textual content poses unique challenges, particularly due to the language's typographical diversity and the scarcity of digital resources. This thesis presents a novel approach to Optical Character Recognition (OCR) tailored to Kazakh book texts, leveraging a synthetic dataset to overcome the limitations of data scarcity and enhance model accuracy. Through meticulous dataset engineering, employing tools like SynthTiger, the study generates images that closely replicate the conditions of Kazakh printed material. The OCR models are rigorously trained and tested, demonstrating high precision in recognizing diverse text presentations. Additionally, this work includes the development of a web application utilizing the EasyOCR framework, which underscores the practical application of the research. Hosted on Hugging Face Spaces, the application offers users the capability to extract text from various image and document formats, illustrating the robustness and adaptability of the OCR models to real-world scenarios.

Аңдатпа

Қазақ мәтіндік контентінің цифрландырылуы ерекше қиындықтар туғызады, әсіресе тілдің типографиялық әртүрлілігі мен сандық ресурстардың жетіспеушілігіне байланысты. Бұл диссертацияда деректердің жетіспеушілігін жеңуге және модельдің дәлдігін арттыруға арналған синтетикалық деректер жиынтығын қолдана отырып, қазақ кітап мәтіндеріне бейімделген жаңа Оптикалық Таңбаларды Танудың (OCR) тәсілі ұсынылады. SynthTiger сияқты құралдарды қолдана отырып, мұқият деректер жиынтығын құрастыру арқылы зерттеу қазақ баспалық материалдарының жағдайларын дәл қайталайтын кескіндер жасайды. OCR модельдері мұқият дайындалып, тексеріліп, әртүрлі мәтіндік презентацияларды тануда жоғары дәлдікті көрсетеді. Сонымен қатар, бұл жұмыста зерттеудің практикалық қолданылуын көрсететін EasyOCR негізіндегі веб-қосымшаның әзірленуі қамтылады. Hugging Face Spaces-те орналастырылған қосымша пайдаланушыларға әртүрлі кескін және құжат форматтарынан мәтінді алу мүмкіндігін ұсынады, OCR модельдерінің нақты әлем жағдайларына төзімділігі мен бейімделгіштігін көрсетеді.

Аннотация

Цифровизация казахского текстового контента представляет уникальные вызовы, особенно из-за типографического разнообразия языка и недостатка цифровых ресурсов. В этой диссертации представлен новый подход к Оптическому Распознаванию Символов (OCR), адаптированный для казахских книжных текстов, с использованием синтетического набора данных для преодоления ограничений дефицита данных и повышения точности модели. Посредством тщательной инженерии набора данных, с применением таких инструментов, как SynthTiger, исследование генерирует изображения, которые точно воспроизводят условия казахского печатного материала. OCR модели тщательно обучаются и тестируются, демонстрируя высокую точность в распознавании различных текстовых презентаций. Кроме того, эта работа включает разработку веб-приложения с использованием фреймворка EasyOCR, которое подчеркивает практическое применение исследования. Размещенное на Hugging Face Spaces, приложение предлагает пользователям возможность извлекать текст из различных форматов изображений и документов, иллюстрируя надежность и адаптивность OCR моделей к реальным условиям.

List of Abbreviations

AI - Artificial Intelligence
BiLSTM - Bidirectional Long Short-Term Memory
CNN - Convolutional Neural Network
CRNN - Convolutional Recurrent Neural Network
CTC - Connectionist Temporal Classification
GAN - Generative Adversarial Network
HMM - Hidden Markov Model
LSTM - Long Short-Term Memory
NED - Normalized Edit Distance
NLP - Natural Language Processing
OCR - Optical Character Recognition
RNN - Recurrent Neural Network
VGG - Visual Geometry Group (a type of neural network)
WRR - Word Recognition Rate

Table of Contents

Chapter 1

Introduction

1.1 Research Background

Optical Character Recognition (OCR) technology has revolutionized the processing of digital information, significantly impacting a wide range of applications, from automated number plate recognition and passport verification to business document management and historical document preservation [1]. By converting images of typed or handwritten text into machine-encoded text, OCR enables seamless transitions from physical documents to digital data management systems. This transformation has been particularly profound over the past few decades, evolving from mechanical systems to sophisticated software solutions driven by advancements in machine learning and deep learning [2].

The progress in OCR has been driven by the advent of deep learning techniques, utilizing large datasets of image-text pairs to train complex neural networks. These advancements have markedly improved text recognition capabilities, extending the applicability of OCR across various domains. However, these benefits have been disproportionately realized by languages with abundant digital resources and annotated data [3]. This uneven distribution underscores the need for targeted research to develop OCR technologies for underrepresented languages.

1.2 Challenges in OCR for Underrepresented Languages

Languages such as Kazakh face significant challenges in OCR integration due to the scarcity of annotated textual resources. This scarcity not only hampers technological development but also threatens cultural preservation by limiting access to important texts in digital formats [4]. The Kazakh language, which uses both Cyrillic and Latin scripts, presents unique difficulties for OCR technology. These challenges are exacerbated by the language's morphological complexity and script variability, which existing datasets, primarily designed for more commonly used languages, do not adequately address [5].

The transition from Cyrillic to Latin scripts in Kazakh further complicates OCR development. OCR systems need to be versatile and robust, capable of accurately

recognizing and processing texts that may contain a mix of scripts. This variability necessitates advanced OCR models that can generalize across different writing systems while maintaining high accuracy.

1.3 Synthetic Datasets: A Solution to Data Scarcity

In response to the limitations imposed by data scarcity, synthetic datasets have emerged as a crucial tool for OCR development. These datasets simulate real-world text appearances, allowing for the robust training of OCR models without requiring large volumes of manually annotated data [6]. Synthetic datasets are not only cost-effective but also scalable, providing a strategic solution to enhance OCR accuracy for languages with minimal existing resources.

Synthetic datasets can be meticulously crafted to include a wide variety of text styles, fonts, and backgrounds, creating a comprehensive training environment for OCR systems. For Kazakh, synthetic datasets can be designed to reflect the linguistic and typographic characteristics of Kazakh texts, significantly improving OCR model performance. This approach allows for the inclusion of diverse and complex text scenarios, such as varying font styles, sizes, and backgrounds typical of Kazakh book texts, which are essential for developing robust and adaptable OCR systems.

1.4 Objectives of This Study

This thesis aims to advance the recognition of Kazakh book texts through the development of a specifically tailored synthetic dataset. The primary objectives of this study are:

- **To create a comprehensive synthetic dataset** that replicates the font styles, sizes, and backgrounds typical of Kazakh book texts.
- **To enhance OCR models** to better address the specific needs of book text recognition, distinguishing it from other forms of text.
- **To evaluate the effectiveness of these models** using the synthetic dataset, aiming to demonstrate significant improvements in the accuracy and efficiency of Kazakh text processing.

1.5 Significance of the Study

The significance of this study lies in its potential to bridge the technological gap in OCR for underrepresented languages. By focusing on Kazakh, this research contributes to the broader effort of preserving linguistic diversity in the digital age. Developing effective OCR systems for Kazakh not only aids in the digital archiving and accessibility of Kazakh literature but also serves as a model for similar initiatives for other underrepresented languages.

The methodologies and insights gained from this study can be applied to enhance OCR technologies globally, promoting inclusivity and ensuring that the benefits of technological advancements are equitably distributed. The success of this

research could lead to improved tools for educators, researchers, and cultural institutions, supporting the preservation and dissemination of linguistic and cultural heritage.

Chapter 2

Literature Review

2.1 Overview of OCR Technologies

Optical Character Recognition (OCR) technology has played a pivotal role in the digitization of textual content, converting scanned images of text into editable and searchable digital formats. The historical development of OCR spans several decades, beginning with mechanical solutions in the mid-20th century and evolving into sophisticated software-based approaches that leverage advanced computational techniques [2, 7].

2.1.1 Early Developments

The early stages of OCR technology focused on mechanical and optical systems that were capable of recognizing text by comparing shapes with hardcoded templates. These systems were highly constrained, typically limited to reading a single font type and size, which severely limited their practical application [2]. These initial efforts laid the groundwork for the conceptual framework of OCR but were far from the flexible, accurate systems we see today.

2.1.2 Advent of Machine Learning

As computational power increased and machine learning techniques became more sophisticated, OCR technologies saw significant advancements. The 1990s marked the beginning of the use of statistical methods, particularly Hidden Markov Models (HMMs), which allowed for more flexible and accurate text recognition by modeling sequences of characters rather than relying on static templates [8]. HMMs brought a statistical approach to OCR, enabling the handling of variations in text appearance and improving the adaptability of OCR systems.

2.1.3 Deep Learning Revolution

The real transformation in OCR came with the application of deep learning, particularly through the use of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), including Long Short-Term Memory networks (LSTMs).

These technologies enabled the development of OCR systems that could learn from a vast array of text samples in various fonts, sizes, and styles, dramatically improving the robustness and accuracy of text recognition [9, 10].

CNNs have been instrumental in feature extraction, capable of detecting intricate patterns in text such as different fonts and distorted text styles found in natural scenes. This capability has been crucial for applications like street sign recognition and information extraction from documents in uncontrolled environments [9].

RNNs, and specifically LSTMs, have been pivotal for modeling the sequence aspect of text, allowing OCR systems to better understand the context within strings of text, which is essential for accurate recognition of sentences and paragraphs. This sequential processing has been particularly beneficial in processing forms and longer documents, where context plays a significant role in interpretation [11].

2.1.4 Integration with Other Technologies

The integration of OCR with technologies such as Natural Language Processing (NLP) and Image Processing has opened up new possibilities for automated document analysis. OCR can serve as a foundational technology for extracting readable text that NLP systems can then analyze and interpret [12]. This integration has enabled the development of more sophisticated applications, such as automatic translation, sentiment analysis, and intelligent document retrieval systems.

2.1.5 Challenges and Future Directions

Despite these advancements, OCR technology still faces challenges, especially when dealing with scripts that are not well-represented in training datasets or in adverse conditions such as poor lighting or low resolution. The robustness of OCR systems in handling variations in text appearance, such as different handwriting styles or unusual fonts, remains an ongoing challenge. Future research directions are likely to focus on improving the adaptability of OCR systems to diverse languages and operational conditions, enhancing the quality of text recognition across the board [13]. Advances in generative models, such as Generative Adversarial Networks (GANs), are expected to play a significant role in creating more realistic synthetic training data, thereby improving OCR performance.

2.2 The Role of Synthetic Datasets in OCR

Synthetic datasets have become an indispensable tool in the field of OCR, particularly with the advent of deep learning technologies. These datasets are engineered to simulate real-world text conditions and are crucial for training OCR systems when authentic annotated data is scarce or difficult to collect [14, 6].

2.2.1 Creation and Application of Synthetic Data

The generation of synthetic text involves creating images of artificial text that mimic the appearance of text in natural environments. This process includes varying the text's font, size, color, and background to reflect the diversity of real-world conditions [14]. SynthText and SynthTIGER are prominent examples of tools that generate such data, offering extensive customization options to enhance the robustness of the trained models [6].

2.2.2 Advantages of Synthetic Datasets

One of the primary advantages of synthetic datasets is their ability to mitigate the challenges associated with the scarcity of labeled data. This is particularly valuable for languages that do not have extensive digital resources or for specialized applications like book text recognition where specific types of text formatting and styles are needed [14]. Creating synthetic datasets is generally more cost-effective and scalable compared to the labor-intensive process of collecting and annotating real-world data. This scalability allows for the rapid expansion of dataset size and diversity, crucial for training deep learning models that require large amounts of data to achieve high accuracy [6].

Synthetic data also offers unparalleled flexibility in training OCR systems. Researchers can easily manipulate data parameters such as text alignment, spacing, and noise level to create challenging scenarios for OCR models. This ability is essential for preparing models to handle adverse conditions they will encounter in real-world applications [10].

2.2.3 Challenges in Using Synthetic Datasets

Despite their advantages, synthetic datasets are not without challenges. The primary concern is the realism gap between synthetic and real images, which can lead to discrepancies in model performance when transitioning from synthetic training environments to real-world applications [13]. Models trained exclusively on synthetic data can overfit to the specific characteristics of the synthetic images, such as unusual fonts or overly clean text layouts, which are not typically encountered in real-world texts [13]. This overfitting can reduce the model's generalization capability on real-world data.

To mitigate these issues, recent research has focused on improving the realism of synthetic datasets through advanced techniques like Generative Adversarial Networks (GANs), which can generate more realistic text images by learning from real data distributions [15]. Additionally, integrating a mix of real and synthetic data during training is recommended to balance the model's exposure to both types of data [6].

2.3 Text Detection Algorithms

Text detection is a critical step in OCR that involves identifying and localizing text regions within an image. This section will discuss notable algorithms such as EAST and Differentiable Binarization.

2.3.1 EAST: An Efficient and Accurate Scene Text Detector

The Efficient and Accurate Scene Text (EAST) detector represents a significant advancement in the domain of optical character recognition (OCR) and scene text detection. Introduced in the influential paper titled "EAST: An Efficient and Accurate Scene Text Detector," this framework is celebrated for its rapid processing speed and high detection accuracy, adeptly addressing common challenges such as variable lighting, diverse orientations, and text scales found in natural scenes.

EAST marks a departure from traditional text detection methodologies by leveraging a fully convolutional network (FCN) that outputs words or text lines directly from full images. This approach eliminates the need for candidate generation and aggregation, typical of many previous methods, thereby simplifying the detection process and significantly enhancing operational speed—key for real-time application use.

At its core, EAST employs an FCN with a U-Net-like architecture. This network features a downsampling path to capture text features at multiple scales and an upsampling path that fuses these features to predict text presence and geometries at each pixel. This dense per-pixel prediction method facilitates direct and accurate text localization and geometry estimation across the entire image, accommodating texts of various shapes and orientations.

EAST introduces a novel loss function that concurrently optimizes for geometry and confidence scoring. This integrated approach ensures precise alignment with actual text geometries—such as rotations and aspect ratio variations—critical for accurate text detection in uncontrolled imaging environments.

EAST streamlines the complex, multi-stage pipelines traditionally employed in text detection into a single, end-to-end trainable model. This simplification not only boosts detection speed but also enhances accuracy, as the model globally optimizes text detection across the entire image. Additionally, EAST's flexibility in processing images of arbitrary dimensions and orientations without resizing or reshaping represents a substantial improvement over less adaptable methods.

Demonstrated through rigorous testing, including the ICDAR challenge benchmarks, EAST has consistently outperformed existing methods in both detection speed and accuracy. Its capability to process images and detect text within fractions of a second per image underscores its suitability for real-time applications, such as mobile OCR, augmented reality interfaces, and robotic navigation, where quick text interpretation is crucial.

The practical implications of EAST's performance are vast, particularly in areas requiring instant textual feedback from the environment. Its use in real-time applications, from mobile apps that translate text on-the-go to interactive educational tools that overlay text information on live images, showcases the broad

utility of this advanced text detection method.

2.3.2 CRAFT: Character Region Awareness for Text Detection

The Character Region Awareness for Text Detection (CRAFT) model represents a significant advancement in the field of scene text detection. Developed by researchers at NAVER Corp's Clova AI Research, CRAFT addresses the limitations of previous methods that relied on rigid word-level bounding boxes by focusing on character-level detection and the affinity between characters. This approach allows CRAFT to effectively detect text in a variety of complex shapes, such as curved, deformed, or arbitrarily oriented text [16].

CRAFT employs a fully convolutional network (FCN) architecture based on VGG-16 with batch normalization. The network outputs two score maps: a character region score, which indicates the presence of individual characters, and an affinity score, which represents the connection or affinity between adjacent characters. This dual-score mechanism enables the model to accurately localize and link characters into coherent text instances, overcoming challenges associated with traditional word-level detection methods [16].

One of the key innovations of CRAFT is its weakly-supervised learning framework. Due to the scarcity of character-level annotations in real datasets, the model is trained using a combination of synthetic data with character-level annotations and real data with word-level annotations. The ground truth for real images is generated through a process involving an interim model that estimates character-level bounding boxes, which are then refined through a series of transformations and algorithms, such as the watershed algorithm. This approach ensures that the model can effectively learn to detect characters even in the absence of explicit character-level annotations in real images [16].

CRAFT has demonstrated state-of-the-art performance across multiple benchmark datasets, including ICDAR 2013, ICDAR 2015, ICDAR 2017, MSRA-TD500, TotalText, and CTW-1500. The model has shown remarkable flexibility and robustness in detecting text in various forms and orientations, significantly outperforming previous methods in handling long, curved, and arbitrarily shaped texts. This capability makes CRAFT particularly well-suited for applications requiring high accuracy in complex text detection scenarios, such as real-time translation, augmented reality, and automated navigation systems for visually impaired users [16].

CRAFT's character-level detection approach provides several advantages over traditional word-level detectors. By focusing on individual characters and their affinities, CRAFT can more accurately detect and represent text regions that are irregularly shaped or oriented. This method reduces the ambiguity associated with word segmentation and allows for more precise text localization, especially in scenes with challenging text layouts. Furthermore, the weakly-supervised training framework enhances the model's ability to generalize across different datasets and text styles, ensuring robust performance in diverse real-world applications [16].

2.3.3 SAST: Single-Shot Arbitrarily-Shaped Text Detector

The Single-Shot Arbitrarily-Shaped Text Detector (SAST) presents a novel approach to detecting scene text of arbitrary shapes. Developed by Pengfei Wang and colleagues, SAST employs a context-attended multi-task learning framework based on a Fully Convolutional Network (FCN) to learn various geometric properties for the reconstruction of polygonal representations of text regions [17].

SAST utilizes a FCN architecture to simultaneously predict multiple geometric properties of text regions, including text center lines (TCL), text border offsets (TBO), text center offsets (TCO), and text vertex offsets (TVO). This multi-task learning formulation enables the model to effectively handle the detection of arbitrarily-shaped text, including curved, multi-oriented, and multilingual text instances.

A key component of the SAST model is the Context Attention Block (CAB), which is designed to capture long-range dependencies of pixel information and enhance feature representation. The CAB integrates contextual information both horizontally and vertically, improving the robustness of the segmentation results. Additionally, the point-to-quad assignment method is introduced in the post-processing stage to cluster pixels into text instances by integrating both high-level object knowledge and low-level pixel information in a single shot. This method helps to accurately reconstruct the polygonal representation of text regions [17].

SAST has demonstrated competitive performance across multiple benchmark datasets, including ICDAR 2015, ICDAR 2017-MLT, SCUT-CTW1500, and Total-Text. It achieves high accuracy and efficiency in detecting arbitrarily-shaped text, running at 27.63 frames per second (FPS) with a high mean (Hmean) of 81.0% on SCUT-CTW1500, surpassing many existing segmentation-based methods. This performance makes SAST suitable for real-time applications, such as augmented reality translation and mobile text recognition [17].

The SAST model offers several advantages over traditional text detection methods. Its single-shot framework eliminates the need for multiple proposal stages, reducing computational redundancy and improving detection speed. The integration of CABs and the point-to-quad assignment method enhances the model's ability to handle complex text layouts and separate closely situated text instances. This approach also addresses the issue of text fragmentation, which is common in long text instances [17].

2.3.4 Differentiable Binarization for Text Detection

Differentiable Binarization (DB) is a novel approach to scene text detection that has gained significant attention due to its ability to effectively handle the challenges associated with the detection of text in natural scenes. The method, introduced in the paper "Real-time Scene Text Detection with Differentiable Binarization," proposes an innovative technique that integrates traditional binarization processes into a deep learning framework, allowing for end-to-end training and real-time performance.

The core idea behind Differentiable Binarization is to streamline the text detection process by using a single neural network that performs both text region

proposal and binarization simultaneously. The network uses a convolutional backbone, typically a variant of the VGG or ResNet architectures, which processes the input image to produce feature maps at multiple scales. These feature maps are then fed into a novel binarization module, which applies a differentiable thresholding mechanism to predict text regions.

The differentiability of the binarization process is the key innovation here, as it allows the gradient of the loss function to pass through the binarization step during backpropagation. This is achieved by replacing the hard binarization step with a differentiable approximation, which smoothly varies the binarization threshold. This setup not only improves the training dynamics by enabling the use of common backpropagation techniques but also enhances the adaptability of the model to different text appearances and conditions.

One of the major advantages of Differentiable Binarization compared to previous scene text detection methods is its robustness to various text characteristics and environmental conditions. By training the network in an end-to-end manner, all components of the model are optimized to work in unison, which significantly enhances the detection accuracy. Furthermore, the real-time capability of the model, as demonstrated in the paper, is particularly important for applications requiring immediate text recognition, such as in augmented reality systems or mobile applications.

The paper presents comprehensive experimental results to validate the effectiveness of the Differentiable Binarization approach. The model is tested against several benchmark datasets commonly used in the text detection field, such as ICDAR 2015, ICDAR 2017 MLT, and Total-Text. The results show that the DB method not only achieves high precision and recall rates but also outperforms existing methods in terms of both speed and accuracy, thereby setting a new state-of-the-art for real-time text detection.

2.4 Text Recognition Algorithms

Text recognition algorithms are responsible for interpreting and converting detected text regions into machine-readable formats. This section will cover various models, including those based on Convolutional Recurrent Neural Networks (CRNN) and other advanced architectures.

2.4.1 CRNN: Convolutional Recurrent Neural Network

The Convolutional Recurrent Neural Network (CRNN) is a novel architecture designed for recognizing sequence-like objects in images, integrating the strengths of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Developed by Baoguang Shi, Xiang Bai, and Cong Yao, CRNN is particularly effective for tasks such as scene text recognition, where it demonstrates significant advantages over traditional methods [18].

CRNN consists of three main components: convolutional layers, recurrent layers, and a transcription layer. The convolutional layers extract a sequential feature representation from the input image. These layers are adapted from a standard

CNN, where the fully connected layers are removed to maintain spatial information and allow the network to process variable-length sequences.

The extracted features are then fed into the recurrent layers, typically composed of bidirectional Long Short-Term Memory (LSTM) networks. These layers capture contextual information within the sequence, leveraging both past and future context to make predictions. The use of bidirectional LSTMs enables the model to effectively utilize information from both directions of the sequence, which is crucial for accurately recognizing characters that may have ambiguous shapes or positions [18].

The transcription layer converts the per-frame predictions from the recurrent layers into a final label sequence. This is achieved using the Connectionist Temporal Classification (CTC) layer, which allows the network to be trained end-to-end without requiring pre-segmented input data. The CTC layer computes the conditional probability of a label sequence given the input sequence, effectively handling variable-length outputs and making the CRNN architecture particularly suited for tasks like scene text recognition where the lengths of sequences can vary significantly [18].

CRNN has shown superior performance on several standard benchmarks for scene text recognition, including the IIIT-5K, Street View Text, and ICDAR datasets. It consistently outperforms traditional methods and other deep learning models by leveraging its end-to-end trainable architecture, which simplifies the pipeline and reduces the need for detailed character-level annotations during training.

The model's ability to directly learn from sequence labels and its robustness to variable sequence lengths make it highly practical for real-world applications. For instance, CRNN can be used in applications such as automatic number plate recognition, document digitization, and real-time translation services where text appears in diverse and unpredictable forms [18].

CRNN offers several advantages over previous text recognition methods. Unlike conventional approaches that rely on separate detection and recognition stages, CRNN provides an integrated framework that is trained end-to-end. This not only simplifies the workflow but also enhances the model's robustness and accuracy by jointly optimizing all components.

Moreover, CRNN eliminates the need for detailed character-level annotations, relying instead on sequence labels that are easier to obtain. This reduces the burden of data annotation and allows the model to generalize better across different datasets and text styles. Additionally, the model's compact architecture, with fewer parameters than standard CNNs, makes it more efficient in terms of storage and computational resources, facilitating deployment on mobile devices and other resource-constrained environments [18].

2.4.2 Rosetta: Large Scale System for Text Detection and Recognition

Rosetta is a scalable OCR system developed by Facebook to process the vast number of images uploaded daily to its platforms, such as Facebook and Instagram.

This system is designed to efficiently detect and recognize text in images, facilitating applications like search and recommendation. Developed by Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar, Rosetta combines advanced modeling techniques with a robust system architecture to handle the high volume and variability of social media images [19].

Rosetta's OCR system operates in two main stages: text detection and text recognition. The text detection stage uses a model based on Faster-RCNN, a state-of-the-art object detection framework. This model employs a fully convolutional network (FCN) to represent images as convolutional feature maps. A region proposal network (RPN) then generates bounding boxes for potential text regions, which are refined and classified as containing text or not. To enhance efficiency, Rosetta replaces the ResNet convolutional body typically used in Faster-RCNN with a ShuffleNet-based architecture, achieving faster processing times without compromising accuracy [19].

In the text recognition stage, Rosetta uses a fully convolutional model that processes the detected text regions to produce character-level transcriptions. Unlike traditional methods that rely on predefined dictionaries, Rosetta's recognition model is lexicon-free, allowing it to recognize words of arbitrary length and content. This model outputs sequences of characters using a Connectionist Temporal Classification (CTC) loss, which aligns the predicted character sequences with the ground truth, accommodating variations in word length and character spacing [19].

Rosetta has demonstrated high performance on various benchmark datasets, including COCO-Text, ICDAR 2015, and ICDAR 2017 MLT. It achieves superior accuracy in detecting and recognizing text across diverse fonts, languages, and image qualities. The system is capable of processing hundreds of millions of images per day in real-time, making it well-suited for large-scale applications on social media platforms.

The practical applications of Rosetta are extensive. By extracting textual information from images, Rosetta supports features such as image search, automatic tagging, and content moderation. It also enhances user experiences by enabling functionalities like real-time translation of text in images and improved accessibility for visually impaired users through text-to-speech conversion [19].

Rosetta offers several advantages over traditional OCR systems. Its use of a fully convolutional, lexicon-free recognition model provides flexibility in recognizing diverse and unpredictable text content, such as URLs, emails, and special symbols. The integration of a ShuffleNet-based architecture for text detection ensures efficient processing, making the system scalable to handle the vast image uploads on social media platforms.

Additionally, Rosetta's end-to-end trainable framework simplifies the OCR, reducing the need for extensive pre- and post-processing steps. This results in faster inference times and improved accuracy, particularly in real-world scenarios with varied image conditions. The system's robust design and deployment at scale demonstrate its capability to meet the demanding requirements of modern social media platforms [19].

2.4.3 SRN: Semantic Reasoning Networks

Semantic Reasoning Networks (SRN) represent a novel approach to scene text recognition, focusing on integrating visual perception with semantic reasoning to enhance recognition accuracy. Developed by Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding, SRN addresses the limitations of previous methods that often relied solely on visual features or employed sequential RNNs for semantic context, which could be inefficient and error-prone [20].

SRN integrates four main components: the backbone network, Parallel Visual Attention Module (PVAM), Global Semantic Reasoning Module (GSRM), and Visual-Semantic Fusion Decoder (VSFD). The backbone network, based on ResNet50 with Feature Pyramid Network (FPN), extracts robust 2D feature maps from the input image. These feature maps are then processed through PVAM, which generates aligned visual features for each character position in a parallel manner, enhancing efficiency and accuracy.

The GSRM plays a crucial role in capturing global semantic context through a multi-way parallel transmission mechanism. Unlike traditional RNN-based methods that rely on sequential processing, GSRM processes all characters simultaneously, which not only improves computational efficiency but also enhances the robustness of semantic context modeling. The semantic features generated by GSRM are then combined with visual features in the VSFD to produce the final character predictions. This fusion is achieved using a gated unit mechanism that dynamically balances the contributions of visual and semantic information [20].

SRN has demonstrated state-of-the-art performance across various benchmark datasets, including ICDAR 2013, ICDAR 2015, IIIT5K, Street View Text (SVT), and CUTE80. It significantly outperforms previous methods, especially in recognizing complex and irregular text. For example, on the ICDAR 2015 dataset, SRN achieves an accuracy of 82.7

The model's ability to handle diverse text forms and its robustness in different contexts make it highly suitable for practical applications such as real-time translation, automatic content moderation, and text extraction from natural scenes. By effectively combining visual and semantic information, SRN provides a comprehensive solution for scene text recognition challenges [20].

SRN offers several advantages over traditional text recognition methods. Its parallel processing framework enhances computational efficiency, making it well-suited for real-time applications. The integration of GSRM allows SRN to capture global semantic context more effectively than RNN-based approaches, reducing the risk of error propagation and improving overall recognition accuracy.

Moreover, SRN's end-to-end trainable architecture simplifies the text recognition pipeline, eliminating the need for separate detection and recognition stages. This streamlined approach not only improves performance but also makes the system easier to deploy and maintain. The model's ability to balance visual and semantic features dynamically ensures that it can adapt to various text recognition scenarios, providing consistent and reliable results [20].

2.4.4 SVTR: Scene Text Recognition with a Single Visual Model

SVTR (Single Visual Text Recognition) represents an innovative approach to scene text recognition, designed to address the limitations of hybrid architectures that combine visual models with sequential modeling. Developed by Yongkun Du and colleagues, SVTR simplifies the recognition pipeline by employing a single visual model for both feature extraction and text recognition within a patch-wise image tokenization framework [21].

SVTR decomposes an image text into small patches named character components, which are processed through hierarchical stages involving component-level mixing, merging, and combining operations. This method employs global and local mixing blocks to capture inter-character and intra-character patterns, leading to a multi-grained character component perception. By focusing on both local component patterns and long-term dependencies among characters, SVTR effectively encodes the stroke-like features and contextual relationships necessary for accurate text recognition.

The model's architecture consists of three stages, each progressively reducing the height dimension of the input patches while maintaining the width. This design choice ensures that SVTR captures multi-scale features and preserves the spatial structure essential for recognizing characters in complex scenes. The final stage involves a combining operation that pools the height dimension, followed by a linear prediction to generate the character sequence.

SVTR has demonstrated state-of-the-art performance on various benchmark datasets, including ICDAR 2013, ICDAR 2015, IIIT5K, and Street View Text (SVT). It achieves high accuracy in recognizing both English and Chinese scene texts, with the SVTR-L (Large) variant significantly outperforming existing methods in terms of recognition accuracy while maintaining efficient inference speed. For instance, SVTR-L achieves 97.2

One of the key advantages of SVTR is its simplicity and efficiency. By utilizing a single visual model, SVTR reduces the computational overhead associated with hybrid architectures that require separate models for feature extraction and sequence modeling. This streamlined approach not only enhances inference speed but also improves the model's ability to generalize across diverse text recognition tasks. Furthermore, the model's architecture is highly scalable, with different variants (SVTR-T, SVTR-S, SVTR-B, and SVTR-L) tailored to meet various performance and resource requirements.

The effectiveness of SVTR is attributed to its ability to learn comprehensive character features through its multi-stage processing framework. The use of global and local mixing blocks ensures that the model captures both fine-grained details and broader contextual information, making it well-suited for practical applications such as real-time text recognition in natural scenes, automatic content moderation, and translation services [21].

2.4.5 ViTSTR: Vision Transformer for Fast and Efficient Scene Text Recognition

ViTSTR (Vision Transformer for Scene Text Recognition) represents a novel approach designed to balance accuracy, speed, and computational efficiency in scene text recognition. Developed by Rowel Atienza, ViTSTR leverages the Vision Transformer (ViT) architecture to create a simple, single-stage model that efficiently processes text images with high accuracy [22].

ViTSTR utilizes the Vision Transformer architecture, which transforms input images into sequences of patches. These patches are then embedded and processed through a series of transformer encoder layers. The key innovation in ViTSTR is the use of a transformer encoder without a separate decoder, which simplifies the architecture and enhances computational efficiency. The input image is divided into non-overlapping patches, each of which is linearly embedded and then passed through the transformer encoder. Positional embeddings are added to the patch embeddings to retain spatial information.

The model architecture is designed to be parameter-efficient and fast. For instance, the ViTSTR-Tiny variant achieves 80.3

ViTSTR's performance is competitive with state-of-the-art models while offering significant improvements in speed and efficiency. For example, ViTSTR-Tiny is 2.4 times faster than the TRBA model while using only 43.4

The model is trained using a combination of synthetic and real-world datasets, including MJSynth (MJ) and SynthText (ST), which provide a diverse set of text images for robust training. Data augmentation techniques are employed to further improve the model's generalization capabilities, particularly for irregular text images that pose significant challenges in real-world scenarios.

ViTSTR demonstrates high performance across various benchmark datasets, including IIIT5K, ICDAR 2013, ICDAR 2015, and SVT. Its ability to handle complex text variations, such as different fonts, orientations, and distortions, makes it a versatile tool for scene text recognition. The model's streamlined architecture and efficient processing enable it to achieve state-of-the-art results with fewer computational resources, making it a valuable contribution to the field of scene text recognition [22].

2.4.6 ABINet++: Autonomous, Bidirectional, and Iterative Language Modeling for Scene Text Spotting

ABINet++ (Autonomous, Bidirectional, and Iterative Language Modeling) introduces a comprehensive approach for scene text spotting by integrating advanced language modeling techniques. Developed by Shancheng Fang, Zhendong Mao, Hongtao Xie, Yuxin Wang, Chenggang Yan, and Yongdong Zhang, ABINet++ aims to overcome the limitations of existing methods by leveraging autonomous learning, bidirectional context, and iterative correction [23].

ABINet++ decouples the text recognizer into a vision model (VM) and a language model (LM), blocking gradient flow between the two to ensure independent learning. This autonomy allows each model to specialize in its task: the VM focuses

on visual feature extraction, while the LM enhances character recognition through contextual understanding. The language model, designed as a Bidirectional Cloze Network (BCN), captures contextual information from both left and right sides of a character sequence, mimicking the cloze task used in human reading.

The iterative correction mechanism in ABINet++ further refines the model's predictions. By repeatedly feeding the output back into the LM, the system progressively corrects errors, improving accuracy. This approach is particularly effective in challenging environments with occluded or noisy text.

ABINet++ achieves state-of-the-art performance across various benchmarks, including scene text recognition and spotting tasks. It excels in handling low-quality images and supports multiple languages, demonstrating superior robustness and adaptability. The model's efficacy is validated through extensive experiments on datasets like ICDAR 2013, ICDAR 2015, IIIT5K, and more, showing significant improvements over previous methods.

Key advantages of ABINet++ include its modular design, which allows for easy integration and enhancement of each component, and its ability to learn from both labeled and unlabeled data through self-training. The inclusion of a Transformer unit within a U-Net architecture further enhances feature representation, particularly for long text sequences, by aggregating horizontal features and integrating position and content attention mechanisms.

Overall, ABINet++ sets a new standard in scene text spotting, combining high accuracy, efficiency, and the capability to generalize across diverse text recognition scenarios [23].

2.4.7 PARSeq: Permuted Autoregressive Sequence Models

Permuted Autoregressive Sequence (PARSeq) models represent an innovative approach to scene text recognition (STR), designed to unify context-free and context-aware inference within a single framework. Developed by Darwin Bautista and Rowel Atienza, PARSeq leverages permutation language modeling (PLM) to learn an ensemble of autoregressive (AR) language models with shared weights, enabling efficient and robust text recognition in diverse conditions [24].

PARSeq uses a Transformer-based architecture, which includes a Vision Transformer (ViT) encoder and a shallow decoder. The encoder extracts features from the input image, which are then processed by the decoder to generate character-level predictions. The model's architecture allows it to process both context-free and context-aware information, making it highly adaptable to various text recognition scenarios.

The key innovation in PARSeq is its use of PLM, which generalizes the autoregressive modeling approach. By training on multiple permutations of the input sequence, PARSeq can leverage bidirectional context, significantly improving its accuracy and robustness compared to traditional AR models. This method addresses the limitations of standard AR models, such as unidirectional bias and inefficiencies in decoding, by allowing the model to consider future tokens during training, thereby enhancing its overall performance.

PARSeq has demonstrated state-of-the-art performance on several benchmark

datasets, including IIIT 5k-word, ICDAR 2013, ICDAR 2015, and others. It achieves high accuracy across various character sets and text orientations, showcasing its robustness in handling real-world images with complex text layouts. For example, PARSeq achieves 96.0

The model's efficiency and accuracy are further enhanced by its ability to perform parallel token processing and iterative refinement. This iterative process allows PARSeq to refine its predictions by considering the entire context of the input image, leading to more accurate and reliable text recognition. This capability is particularly useful in applications requiring real-time text recognition, such as augmented reality, automated content moderation, and accessibility tools for visually impaired users [24].

PARSeq offers several advantages over previous STR methods. Its unified structure simplifies the text recognition pipeline, reducing the need for separate models or stages. This not only improves computational efficiency but also enhances the model's ability to generalize across different datasets and text types. Additionally, the extensive use of attention mechanisms within the model ensures that it can handle arbitrarily oriented and occluded text, common challenges in real-world STR tasks.

Chapter 3

Dataset Creation

This chapter describes the development of a synthetic dataset tailored for the optical character recognition (OCR) of Kazakh book texts. Given the scarcity of annotated resources for the Kazakh language, the creation of a synthetic dataset is imperative to advance OCR technologies in this domain.

Kazakh, as an underrepresented language in digital resources, faces significant challenges in the development of effective OCR systems. The limited availability of annotated text images hampers the development of models capable of handling the language's complex typographical styles and structured formats typical in book contexts. Synthetic datasets provide a crucial workaround, enabling the exploration and learning of textual patterns not available in existing datasets.

3.1 Text Source Preparation

The comprehensive preparation of the text source forms the crux of the synthetic dataset creation for optical character recognition (OCR) systems. This foundational process is designed to emulate the complexities and diversities of real-world textual environments that are typical in both Kazakh and English book texts. A meticulously prepared text source not only enhances the robustness of the OCR models but also significantly improves their adaptability and accuracy across various textual formats and styles.

The corpus generation begins by defining a broad set of characters, which includes an array of symbols that are essential for a realistic simulation of textual content. This set encompasses basic punctuation marks such as "!?.,:;" to simulate sentence endings and pauses, numeric digits from "0" to "9", and an assortment of special characters like "'#()<>+-/*=%$»«". Moreover, the inclusion of space symbols alongside a comprehensive list of Kazakh and English letters ensures that the dataset can support the OCR system's ability to process texts from bilingual or multilingual documents, reflecting the linguistic diversity encountered in typical usage scenarios.

Following the character set definition, the process of assembling the text corpus involves the integration of words from two key sources: 'kk_dict.txt' for Kazakh and 'mjsynth.txt' for English. This dual-source approach ensures a rich linguistic mixture, crucial for training sophisticated OCR models. Each word is carefully

screened to ensure it contains only the previously defined characters, thereby maintaining the consistency and purity of the training data.

The dynamic construction of text sequences from these words is guided by specific rules designed to reflect the natural variability found in book texts. Each sequence is generated to not exceed 25 characters, mirroring the average line length in printed books. To build these sequences, the script randomly concatenates words and intersperses them with symbols, simulating the syntactic and typographic diversity typical of book content. This includes not only straightforward word lists but also the insertion of mathematical expressions, formatted dates, and numerically annotated lists, which are typical in academic and non-fiction texts.

To further enrich the dataset, special functions within the script generate complex data structures: - Mathematical expressions are crafted by randomly combining numbers with basic arithmetic operators and relational symbols, reflecting the usage in technical or scientific materials. - Date and time expressions are generated to reflect a wide temporal range, from historical texts to contemporary writings, making the dataset applicable across various historical contexts. - Numerical data is formatted in diverse ways, including currency and percentage formats, to mimic financial and statistical reports.

In addition to these features, the script introduces randomness in text capitalization to mimic the typographic nuances of real-world texts, where titles and proper nouns are often capitalized. This variability is crucial for training OCR models to recognize and accurately interpret the beginning of sentences and proper names.

Each generated sequence is then meticulously logged into the corpus file, which serves as a vital resource for training the OCR models. This file not only captures the linguistic and typographic diversity but also includes metadata about the generation process, which can be used for subsequent analysis and refinement of the models.

By detailing every aspect of the text source preparation process, from character set definition to the nuanced generation of text sequences, this approach ensures that the synthetic dataset is not only comprehensive and diverse but also finely tuned to meet the specific challenges posed by Kazakh and English OCR tasks. This meticulous preparation is instrumental in advancing OCR technology, particularly in enhancing its performance and reliability in processing complex and varied text environments.

3.2 Detailed Configuration of SynthTiger Components

The SynthTiger tool utilizes a comprehensive set of components designed to generate synthetic images that closely mimic the typographical and aesthetic characteristics of printed book texts. Each component has a specific role in the overall image generation process, and they are finely tuned to produce a variety of text scenarios that an OCR model might encounter.

Corpus and Text Management The corpus component is pivotal as it directs the selection of text that will be rendered onto the images. It is configured to randomly select strings from a specified source, which contains a mix of Kazakh and English phrases prepared earlier:

- **Text Source:** Specified via the `paths` parameter, pointing to the text files used as the source for text generation.
- **Text Length Control:** The `min_length` and `max_length` settings ensure that the text snippets used are between 1 and 25 characters long, reflecting the variability in word and phrase length in actual book texts.

Font Style and Size Adaptation The font component dynamically selects typefaces from a predefined directory (`resources/kz_fonts`), which contains a variety of fonts that support both Latin and Cyrillic scripts, essential for displaying Kazakh and English. This component is crucial for simulating the diverse typographic styles found in books:

- **Font Size Range:** Configured to vary between 10 and 40 points, this range allows the simulation of various text sizes typically found in book publications, from footnotes to headings.
- **Font Style Variation:** While the boldness parameter is set to 0 indicating no bold fonts, this setup could be adjusted in further iterations to include bold and italic styles, adding to the realism of the generated texts.

Background and Texture Configuration Background textures are fundamental in creating a realistic environment for the text. The texture component selects from an array of image files that simulate different paper types and book cover materials:

- **Texture Source and Variability:** The `paths` parameter points to a collection of texture images (`resources/book_texture`) that mimic various paper qualities, from smooth to coarse grains.
- **Transparency and Cropping:** Textures are applied with varying degrees of transparency (`alpha` settings from 0 to 1) and are sometimes cropped (`crop` set to 1) to fit the text layout, enhancing the visual depth and complexity of the background.

Color and Style Dynamics The manipulation of color and style is handled through several sub-components that adjust the visual appearance of both text and background:

- **Color Maps and Gradients:** The `colormap2` and `colormap3` components provide mechanisms to apply bi- or tri-tonal color schemes to the text, potentially simulating color fading effects that are common in older printed materials.
- **Simple and Complex Color Applications:** The `color` component can apply a uniform color or a gradient, and its settings (`gray`, `alpha`, `colorize`) determine the grayscale level and opacity of the text and background, mimicking the ink variations found in real-world scenarios.

Transformations and Layout Adjustments Text placement and orientation are crucial for authenticity, and are managed by the `layout` and `transform` components:

- **Geometric Distortions:** Text may undergo various transformations such as skewing (`Skew`), rotation (`Rotate`), and perspective adjustments (`Perspective`, `Trapezoidate`), which help simulate the physical deformations that occur in bound books.
- **Layout Styles:** Text can be arranged in a flowing line (`FlowLayout`) or along a curve (`CurveLayout`), with settings to adjust the space between characters and the curvature degree, reflecting the stylistic layout choices employed in different types of books.

Post-Processing Effects Once the text and backgrounds are combined, several post-processing effects are applied to integrate the elements more seamlessly and to introduce common imaging artifacts found in scanned documents:

- **Noise and Blur:** Additive Gaussian noise (`AdditiveGaussianNoise`) and Gaussian blur (`GaussianBlur`) are applied to simulate the scanning imperfections and ink spread.
- **Image Resampling:** The `Resample` component occasionally resizes images, simulating the effects of image compression and resolution changes during typical document scanning processes.

Through these detailed configurations and the dynamic interplay of various components, the SynthTiger framework generates images that not only visually represent the text scenarios one might find in Kazakh and English books but also provide challenging data for training robust OCR models. This sophisticated simulation process is crucial for developing OCR technology capable of accurately processing the rich textual heritage of Kazakhstan, thereby supporting its preservation and accessibility in the digital age.

3.3 Label Studio for testing Dataset Creation

Label Studio is an open-source data labeling tool that enables researchers and developers to create labeled data for machine learning applications efficiently. In our project, we utilized Label Studio to create a testing dataset specifically designed to evaluate the performance of our OCR models for Kazakh text recognition.

The flexibility of Label Studio allowed for the customization of the labeling interface to suit the specific needs of our text recognition tasks. Annotations in Label Studio can range from simple classifications to complex layered annotations necessary for detailed textual analysis. This versatility was crucial for annotating various typographical features present in Kazakh texts, which are often challenging due to the script's complexity.

The annotation process involved multiple annotators who were trained to identify and label several text attributes accurately. These attributes included font types, sizes, and text orientations, which are critical for the robustness of OCR models. Label Studio's collaborative tools supported this multi-annotator setup efficiently, enabling real-time updates and consistency checks to ensure high-quality

data annotations.

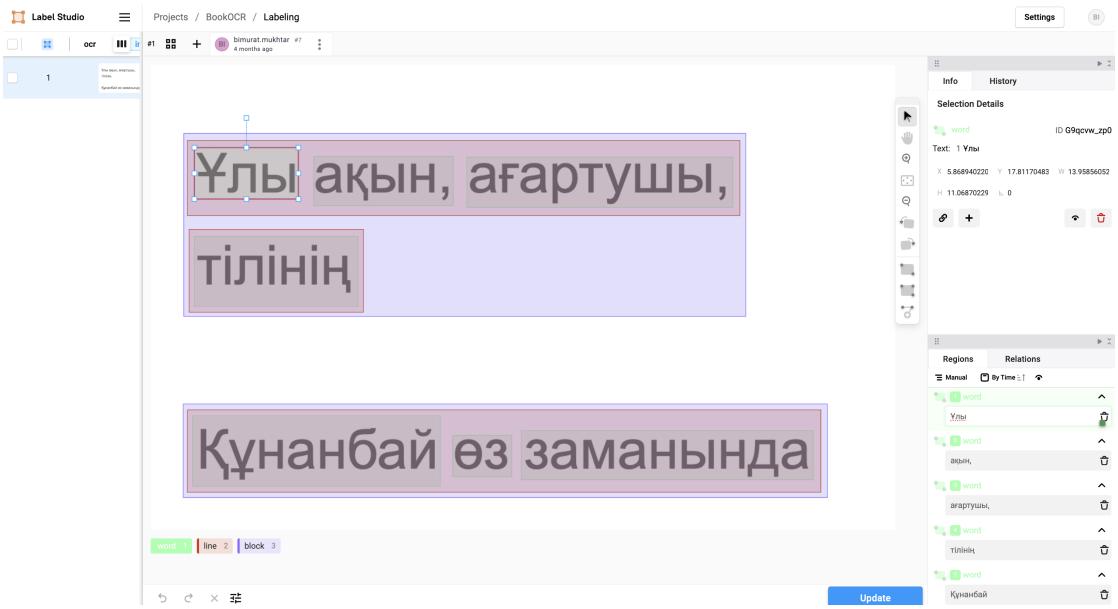

Figure 3.1 - Label studio interface for labeling testing OCR dataset

Custom Machine Learning Backend Integration Alongside Label Studio, we developed a custom ML backend designed to interact seamlessly with the labeled data. This backend utilized several machine learning algorithms optimized for text recognition tasks, particularly focusing on the challenges posed by the Kazakh language's orthographic characteristics.

The backend architecture was built on a modular design, allowing for easy integration of various machine learning models and scalability to accommodate additional functionalities as needed. This modular nature also facilitated the iterative testing and refinement of models based on the annotated data provided by Label Studio.

One of the key features of our custom ML backend was its capability to process and analyze the labeled data dynamically. It included automated pipelines for model training, validation, and testing, which were crucial for the continuous improvement of the OCR accuracy. The backend was also equipped with advanced analytics to monitor model performance and generate detailed reports on the effectiveness of different model configurations.

In conclusion, the combination of Label Studio for dataset creation and a custom ML backend for processing and model integration formed a comprehensive approach to developing a robust OCR system for Kazakh text. This methodology not only streamlined the workflow but also enhanced the accuracy and efficiency of our text recognition models, demonstrating significant improvements in recognizing the diverse typographical features of Kazakh scripts.

Chapter 4

Model Training

Training robust Optical Character Recognition (OCR) models for Kazakh book text recognition involves a comprehensive setup that includes dataset preparation, model selection, and iterative training and testing. This section outlines the methodologies employed in the training process using the synthetic dataset created as described in previous chapters.

4.1 Integration of EasyOCR Framework in OCR System Design

Optical Character Recognition (OCR) has advanced significantly with the development of machine learning techniques, particularly deep learning. The EasyOCR framework, as an accessible and powerful tool, embodies these advancements, offering a modular approach to OCR that can be readily integrated into various applications. This section delves into the EasyOCR framework's architecture, its application within the OCR system designed for Kazakh text recognition, and its harmonization with the other model components outlined in this paper.

4.1.1 Overview of EasyOCR

EasyOCR is a contemporary OCR library that offers a pre-trained model capable of recognizing characters from multiple languages with support for script detection. It is an open-source project that stands out due to its ease of use and ability to be fine-tuned for specific OCR tasks, such as recognizing Kazakh text in natural images.

As shown in Figure 4.1, the framework utilizes a pipeline beginning with pre-processing, followed by text detection using the CRAFT model, mid-processing, recognition via a CRNN that combines ResNet, LSTM, and CTC loss, and concludes with decoding and post-processing to yield the final output.

4.1.2 Pre-processing Stage

The pre-processing stage is critical for preparing images for text detection and recognition. This stage involves normalization, binarization, and possibly denois-

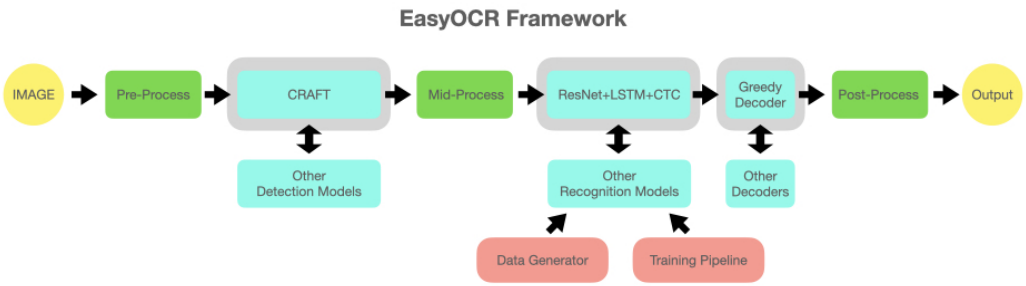

Figure 4.1 - The EasyOCR Framework, illustrating the pipeline from image input to final text output, inclusive of the CRNN components. (reproduced from [25])

ing to ensure that the input images are in a form that maximizes the efficiency of the text detection module. For Kazakh texts, specific pre-processing steps, such as handling cursive scripts and managing typography variations, are crucial.

4.1.3 Decoding Strategies

EasyOCR uses a greedy decoder to translate the predictions from the LSTM network into text. While greedy decoding is straightforward, the framework also provides the flexibility to integrate more sophisticated decoding algorithms, which may offer improved accuracy for complex recognition tasks or when dealing with large lexicons.

4.1.4 Post-processing

In the final stage, the post-processing module refines the raw output from the decoder. This can involve correcting common recognition errors, adjusting character spacing, and enhancing the readability of the output text.

4.1.5 Integration with the Proposed OCR System

The modular design of the EasyOCR framework aligns with the OCR system proposed in this paper for recognizing Kazakh text. The system's architecture leverages EasyOCR's components, particularly its robust text detection and recognition modules, while also allowing for custom enhancements tailored to the Kazakh language.

For instance, the flexibility to incorporate other detection models or decoders within the EasyOCR framework facilitates the adaptation to the unique challenges presented by Kazakh texts. Additionally, the training pipeline of EasyOCR can be fine-tuned with a data generator designed to create synthetic data that closely mimics the intricacies of Kazakh script.

Model Architecture: The architecture of the OCR system is designed to handle the complexities associated with the diverse typography and script variations present in Kazakh texts. The system integrates two main components: a text detection module and a text recognition module.

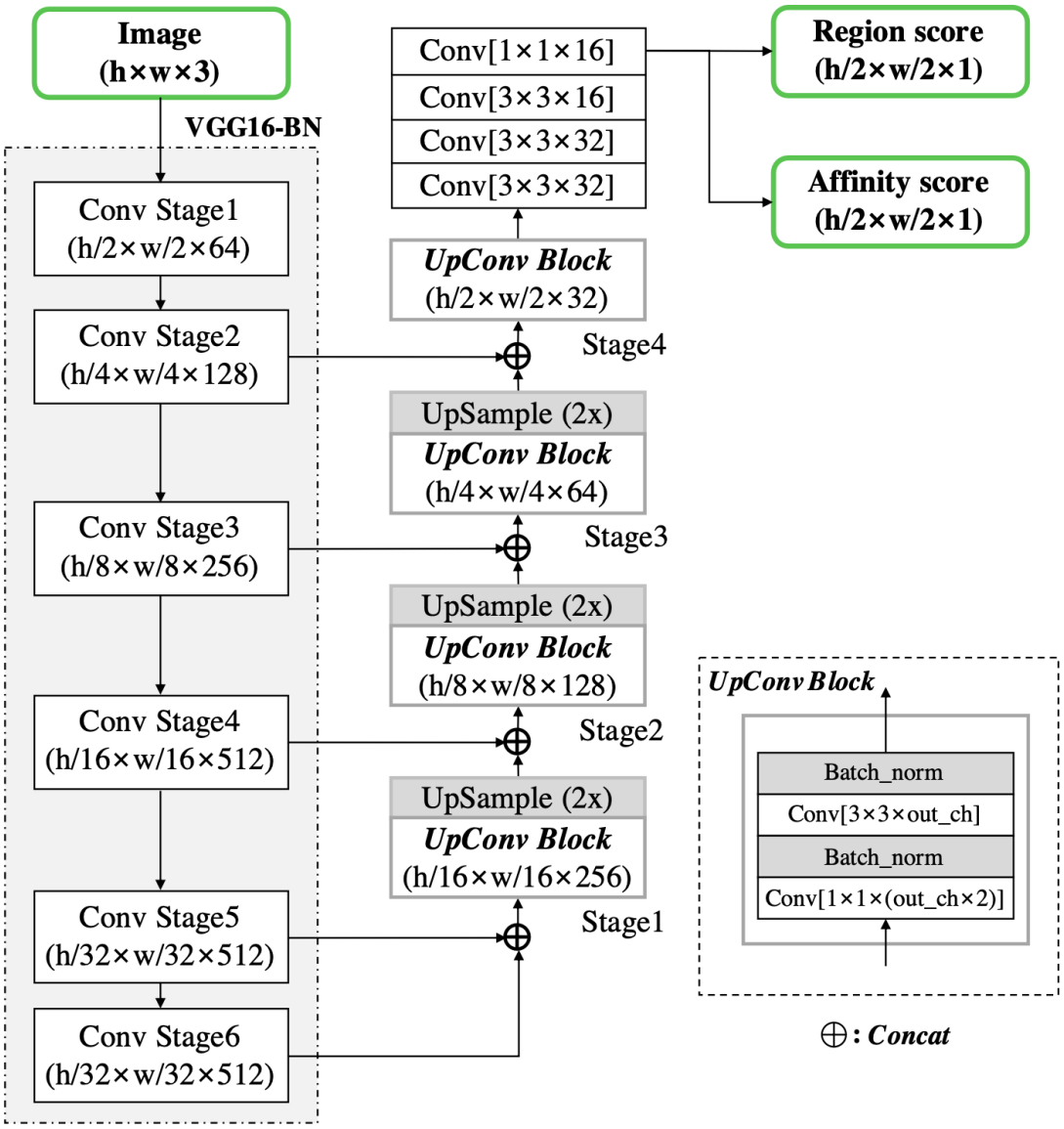

Figure 4.2 - The CRAFT model architecture (reproduced from [26]).

4.1.6 Text Detection

The text detection module employs the CRAFT (Character Region Awareness for Text) model, which is renowned for its ability to accurately detect text regions within images, even against complex backgrounds and varied text orientations. CRAFT uses a convolutional neural network to generate character and affinity scores which indicate, respectively, the likelihood of character presence and the degree of connection between characters. This model excels in detecting text in challenging layouts such as curved or arbitrarily oriented texts by effectively linking character regions in a bottom-up manner.

CRAFT's architecture involves multiple convolutional stages that refine the detection granularity from initial coarse identification of text areas to precise character and affinity scoring. This is achieved through an up-convolution process that magnifies feature maps to higher resolutions, enabling finer detail in detection outputs.

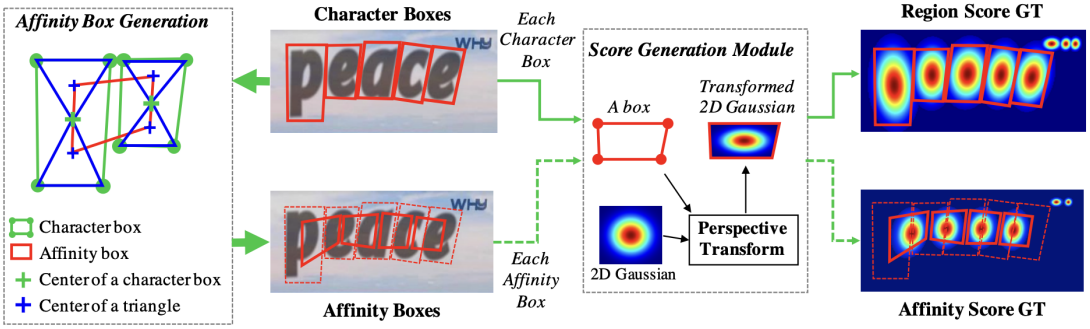

Figure 4.3 - Illustration of ground truth generation procedure in the CRAFT framework (adapted from Baek et al. [26]).

4.1.7 Text Recognition Module

The text recognition module is a fusion of several advanced neural network architectures, each contributing to the model's ability to interpret complex text images. This segment details each component's functionality and interplay within the system.

- **VGG Neural Networks:** The architecture employs the principles of VGG networks, as introduced by Simonyan and Zisserman in their influential work [27]. The design choice of using 3×3 convolutional filters allows for the construction of deeper networks by reducing the number of parameters and improving training times. In the context of OCR, VGG layers serve to distill textual features from the raw image, which are instrumental for the recognition process [27].
- **Bidirectional LSTM (BiLSTM):** BiLSTM networks, extending the capabilities of traditional LSTMs, have proven effective in sequential data modeling, as they capture information from both past and future states [28]. In OCR systems, they are particularly useful for understanding the context in

which characters appear, which is essential for scripts that exhibit significant variability and cursive connections [28].

- **Connectionist Temporal Classification (CTC) Loss:** The CTC loss function, introduced by Graves et al., is a method for training RNNs to map sequences of input data to output labels [29]. Unlike traditional loss functions, CTC does not require pre-aligned data, making it ideal for tasks like speech and handwriting recognition where alignment can be challenging. In OCR, it facilitates the direct prediction of the text sequence from the image, accommodating the varying lengths of textual content without segmentation [29].

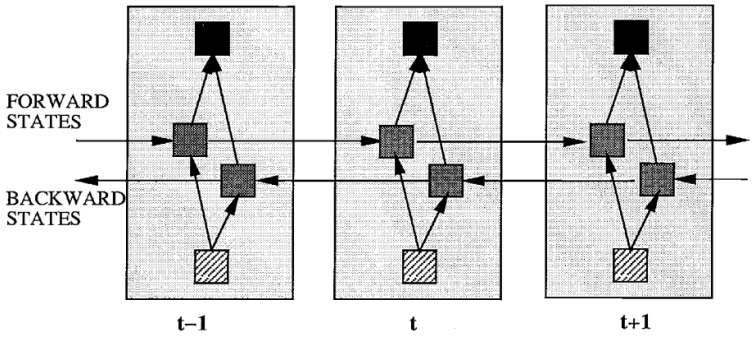

Figure 4.4 - BiLSTM network (image adapted from [30]).

Each of these components is integral to the text recognition process, creating a robust system capable of interpreting a wide variety of textual inputs. The deep feature extraction of VGG, contextual awareness of BiLSTM, and sequence decoding capability of CTC collectively handle the intricacies of Kazakh text recognition with high accuracy.

Deep Dive into the CRNN Architecture: The CRNN is an innovative architecture that brings the potent feature extraction capabilities of CNNs and the sequence modeling proficiency of RNNs into a single, unified model. It is particularly designed for image-based sequence recognition tasks, such as OCR, where it demonstrates remarkable effectiveness.

- **Feature Extraction with CNN:** The CNN component in CRNN, inspired by the VGG-VeryDeep models [27], consists of multiple convolutional and max-pooling layers. These layers convert the input image into a sequence of feature maps. Importantly, fully-connected layers are excluded, leading to a compact representation. The convolutional layers capture high-level features from the input image and translate them into a sequential form, apt for the recurrent layers that follow [31].
- **Sequence Modeling with RNN:** On top of the CNN, the RNN, particularly Long Short-Term Memory (LSTM) networks, predicts a label for each time step in the sequence. LSTMs are adept at handling long-term dependencies, a common occurrence in sequence prediction tasks. The bidirectional LSTM layers enable the model to incorporate context from both directions, enhancing the recognition capability for text, where forward and backward context is equally important [28, 29].

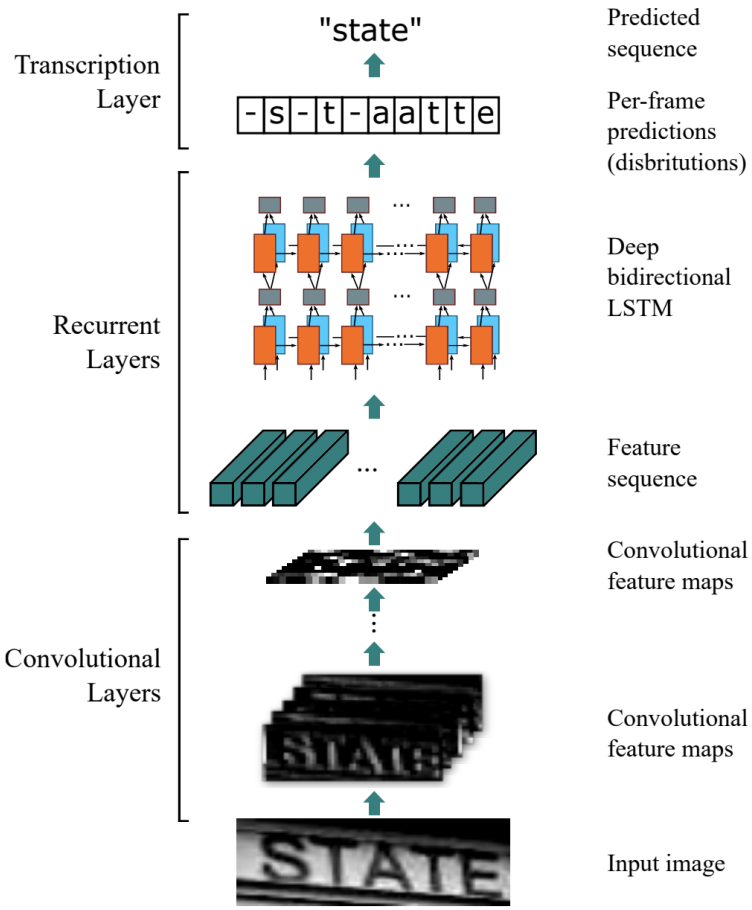

Figure 4.5 - An overview of the CRNN model showing the flow from the input image to the predicted sequence. The architecture encompasses convolutional layers for feature extraction, recurrent layers for sequence modeling, and a transcription layer for sequence prediction (image adapted from [31]).

- **Transcription Layer:** The transcription layer converts the per-frame predictions into a label sequence. This process utilizes the Connectionist Temporal Classification (CTC) algorithm, which aligns the input sequences with their labels efficiently without explicit segmentation [29]. The CTC loss is pivotal for training the CRNN, as it allows the network to be trained end-to-end on unsegmented data, a significant advantage over traditional models which require fine-grained annotations.

The CRNN architecture's end-to-end trainability is one of its core strengths, allowing direct learning from image data to sequence labels without the need for intermediate steps. It eliminates the necessity for manual feature engineering and allows the model to leverage raw image data to its fullest potential [31]. Experiments have demonstrated that the CRNN outperforms traditional methods and other neural network-based approaches in a variety of text recognition tasks [31].

4.2 Training Process

The design of an OCR system capable of interpreting the intricate scripts of the Kazakh language requires a meticulous training process that accommodates the complexity of the script and the variability of real-world application scenarios. This section outlines the comprehensive training process adopted for the EasyOCR framework, from dataset preparation to parameter optimization, and the strategies implemented to overcome challenges encountered during training.

4.2.1 Dataset Preparation and Augmentation

The foundation of an effective OCR system lies in the comprehensiveness and quality of its training data. For the development of the OCR system catered to the Kazakh language, a substantial synthetic dataset comprising over half a million images was curated. This dataset was engineered to simulate an array of textual characteristics observed in Kazakh texts, with variations in font styles, sizes, and intricate backgrounds.

To ensure the diversity of the dataset, an extensive library of fonts representing traditional and modern Kazakh typography was utilized. The font sizes ranged from small to large, accommodating the varying dimensions of text encountered in books. Background augmentation played a crucial role in preparing the model for real-world conditions, with a spectrum of noises, colors, and textures incorporated to represent the challenges posed by non-uniform backgrounds.

4.2.2 Model Configuration

The architecture and configuration of the OCR model directly influence its capability to learn and generalize from the training data. The EasyOCR framework provides a robust foundation for this task, featuring a CRNN architecture. A meticulous configuration process was undertaken to adapt the model to the intricacies of the Kazakh language.

Model Architecture: The architecture employed in the training process is a fusion of CNN and RNN models. The CNN, based on the VGG architecture, is employed for feature extraction, where it excels at identifying textural and spatial patterns within the image data. Following feature extraction, the BiLSTM model – a type of RNN that captures bidirectional dependencies within sequences – processes the sequential data for character recognition. Finally, the CTC loss function aligns the predictions with the target labels, crucial for sequence learning tasks.

- **Batch Size and Iterations:** The training was conducted using a batch size of 96 images. This size was chosen to optimize the trade-off between memory usage and the model's ability to generalize from the data. The number of iterations was set to 100,000 to ensure thorough learning without overfitting.
- **Optimizer and Learning Rate:** Adam optimizer with a learning rate of 1.0 was used. The beta1 parameter, which controls the decay rate of the first moment estimate, was set to 0.9. The rho parameter for Adadelta, which governs the decay rate of the squared gradient moving average, was set to 0.95, with eps providing numerical stability during optimization.
- **Grad Clip:** The gradient clipping was utilized with a threshold of 5 to prevent exploding gradients, a common issue in RNNs that can derail the training process.

4.2.3 Training Execution

The training execution involved iterative forward and backward passes where the model learned to minimize the CTC loss, thereby improving its character and word recognition rates. Each iteration consisted of the following steps:

1. **Forward Pass:** Each image in the batch is passed through the network, producing a sequence of predictions for each segment of the image.
2. **Loss Calculation:** The CTC loss is computed by comparing the predicted sequence against the ground truth labels.
3. **Backward Pass:** Gradients of the loss are calculated with respect to each model parameter.
4. **Parameters Update:** The model parameters are updated using the gradients, guided by the Adam optimization algorithm.

Regular checkpoints were made to assess the model's performance on a validation set. These checkpoints also served as a mechanism to prevent loss of progress and facilitate model tuning without restarting the training from scratch.

4.2.4 Performance Metrics and Evaluation

Performance metrics are crucial for evaluating the effectiveness of the training process. The Character Recognition Rate (CRR), Word Recognition Rate (WRR), and Normalized Edit Distance (NED) served as the primary metrics for this assessment.

Word Recognition Rate (WRR): While CRR focuses on individual characters, WRR assesses the model's accuracy at the word level, which is more indicative of real-world utility. For scripts like Kazakh, where contextual understanding is

paramount, WRR provides insight into how well the model reads continuous text, an essential aspect of OCR functionality.

Normalized Edit Distance (NED): NED calculates the minimum number of edits needed to convert the model's output into the correct text, normalized over the length of that text. It is a comprehensive measure that accounts for insertions, deletions, and substitutions, offering a granular view of the model's performance.

4.2.5 Training Challenges and Solutions

The training journey was interspersed with unique challenges, each necessitating tailored solutions to enhance model performance.

Diverse Fonts and Scripts: The Kazakh language, rich in script variability, presented initial difficulties. To combat this, the training set was diversified with a broader selection of fonts, particularly targeting the more intricate and less commonly used styles to bolster the model's versatility.

Background Complexity: Early models exhibited difficulties discerning text from complex backgrounds. A strategic enhancement in the data generation process introduced a wider range of background noise patterns and improved text-background contrast, leading to a more robust model capable of handling diverse visual scenarios.

Variable Text Orientation: Kazakh texts, especially in artistic depictions, may feature variable orientations. To address this, training data included text at various angles, and the model architecture was adjusted to better detect and interpret rotated or skewed text.

4.2.6 Optimization Strategies

Beyond addressing challenges, specific optimization strategies were employed to refine the training process.

Hyperparameter Tuning: The model's hyperparameters, including batch size, learning rate, and optimizer settings, were tuned iteratively. This fine-tuning was crucial in navigating the trade-offs between learning efficiency and model convergence.

Regularization Techniques: Dropout and batch normalization were applied strategically within the network to combat overfitting. Regularization ensured that the model's learning was generalized and not overly tailored to the training data.

Gradient Clipping: To mitigate the issue of exploding gradients, a common challenge with LSTM networks, gradient clipping was implemented. This technique prevents the gradients from becoming too large and destabilizing the learning process.

4.2.7 Continuous Monitoring and Evaluation

Throughout the training process, continuous monitoring was essential to ensure model improvement was on the right trajectory.

Validation Checks: Periodic evaluations on a held-out validation set provided immediate feedback on the model's generalization capabilities. These checks were instrumental in making timely adjustments to the training regimen.

Loss Tracking: Monitoring the CTC loss over iterations offered insights into the learning progression. Anomalies in loss trends triggered investigations into potential issues such as data quality or network architecture concerns.

Performance Plateaus: Encountering plateaus in metric improvement led to exploring advanced techniques like learning rate annealing or incorporating additional synthetic data to stimulate further learning.

Chapter 5

Results and Discussion

This chapter evaluates the outcomes from the training of Optical Character Recognition (OCR) models on a synthetic dataset created specifically for Kazakh book text recognition. It reflects on the effectiveness of the synthetic dataset, assesses the model's performance using key metrics, and discusses the broader implications of these results.

5.1 Model Performance Evaluation

The OCR models underwent extensive testing to measure their effectiveness. The evaluation was anchored on several performance metrics critical for assessing the practical utility of the models in real-world applications. The models exhibited a low training loss (CTC Loss) of 0.04451 and a validation loss of 0.04288, suggesting effective learning and generalization capabilities. Furthermore, they achieved an impressive accuracy of 92.152% and a normalized edit distance (Norm_ED) of 0.9874, indicating high precision in text recognition and the ability to accurately transcribe Kazakh book text.

	IIIT5k			**SVT**		**IC03**				**IC13**
	50	**1k**	**None**	**50**	**None**	**50**	**Full**	**50k**	**None**	**None**
ABBYY [34]	24.3	-	-	35.0	-	56.0	55.0	-	-	-
Wang *et al.* [34]	-	-	-	57.0	-	76.0	62.0	-	-	-
Mishra *et al.* [28]	64.1	57.5	-	73.2	-	81.8	67.8	-	-	-
Wang *et al.* [35]	-	-	-	70.0	-	90.0	84.0	-	-	-
Goel *et al.* [13]	-	-	-	77.3	-	89.7	-	-	-	-
Bissacco *et al.* [8]	-	-	-	90.4	78.0	-	-	-	-	87.6
Alsharif and Pineau [6]	-	-	-	74.3	-	93.1	88.6	85.1	-	-
Almazán *et al.* [5]	91.2	82.1	-	89.2	-	-	-	-	-	-
Yao *et al.* [36]	80.2	69.3	-	75.9	-	88.5	80.3	-	-	-
Rodrguez-Serrano *et al.* [30]	76.1	57.4	-	70.0	-	-	-	-	-	-
Jaderberg *et al.* [23]	-	-	-	86.1	-	96.2	91.5	-	-	-
Su and Lu [33]	-	-	-	83.0	-	92.0	82.0	-	-	-
Gordo [14]	93.3	86.6	-	91.8	-	-	-	-	-	-
Jaderberg *et al.* [22]	97.1	92.7	-	95.4	80.7*	**98.7**	**98.6**	93.3	**93.1***	**90.8***
Jaderberg *et al.* [21]	95.5	89.6	-	93.2	71.7	97.8	97.0	93.4	89.6	81.8
CRNN	**97.6**	**94.4**	**78.2**	**96.4**	**80.8**	**98.7**	97.6	**95.5**	89.4	86.7

Figure 5.1 - CRNN model performance (image adapted from [18])

In addition to the quantitative metrics, the OCR system underwent a comprehensive qualitative evaluation. This phase involved subjecting the system to a series of tests using scanned pages from Kazakh books, which were deliberately excluded from the training dataset. These pages represented a diverse array of conditions, featuring texts of different sizes and subjected to varying environmental influences that one might encounter in typical book formats. The tests confirmed the system's robustness and adaptability, illustrating its capability to effectively process and recognize text across a multitude of presentation styles. Below, we present several examples of test images and discuss the OCR system's performance in each scenario.

5.2 Practical Application and Web Interface Development

The practical utility of our OCR system was further demonstrated through the development of a user-friendly web application, hosted on Hugging Face Spaces. This application harnesses the EasyOCR framework to provide end-users with the ability to extract text from images and PDFs uploaded directly to the interface.

The application was developed using Streamlit, a powerful and agile web framework that enables rapid prototyping and deployment of data applications. Users can upload images or PDFs, and the system, backed by the EasyOCR engine, processes these files to extract textual content. The interactive platform allows users to test the OCR system's capabilities in real-time with their documents, reflecting the system's versatility in handling different file formats and document conditions.

To facilitate a comprehensive evaluation, images of varying sizes and conditions were tested, with several examples showcased in the web application. The test images included pages from Kazakh books featuring different fonts, layouts, and background complexities, such as the image depicted below. These images were carefully chosen to represent the diverse challenges that the OCR system can encounter in practical applications.

5.2.1 Web Application Functionalities

The web application includes features that allow for the download of extracted text in multiple formats. Users can download the results in plain text or DOCX format, depending on their needs. The code snippets below outline the functionality enabling these features:

```
# download link for the extracted text in TXT format
def generateTxtLink(result): ...

# download link for the extracted text in DOCX format
def generateDocLink(result): ...

# Displaying the image and processing the file uploaded by the user
if uploaded_file is not None:
```

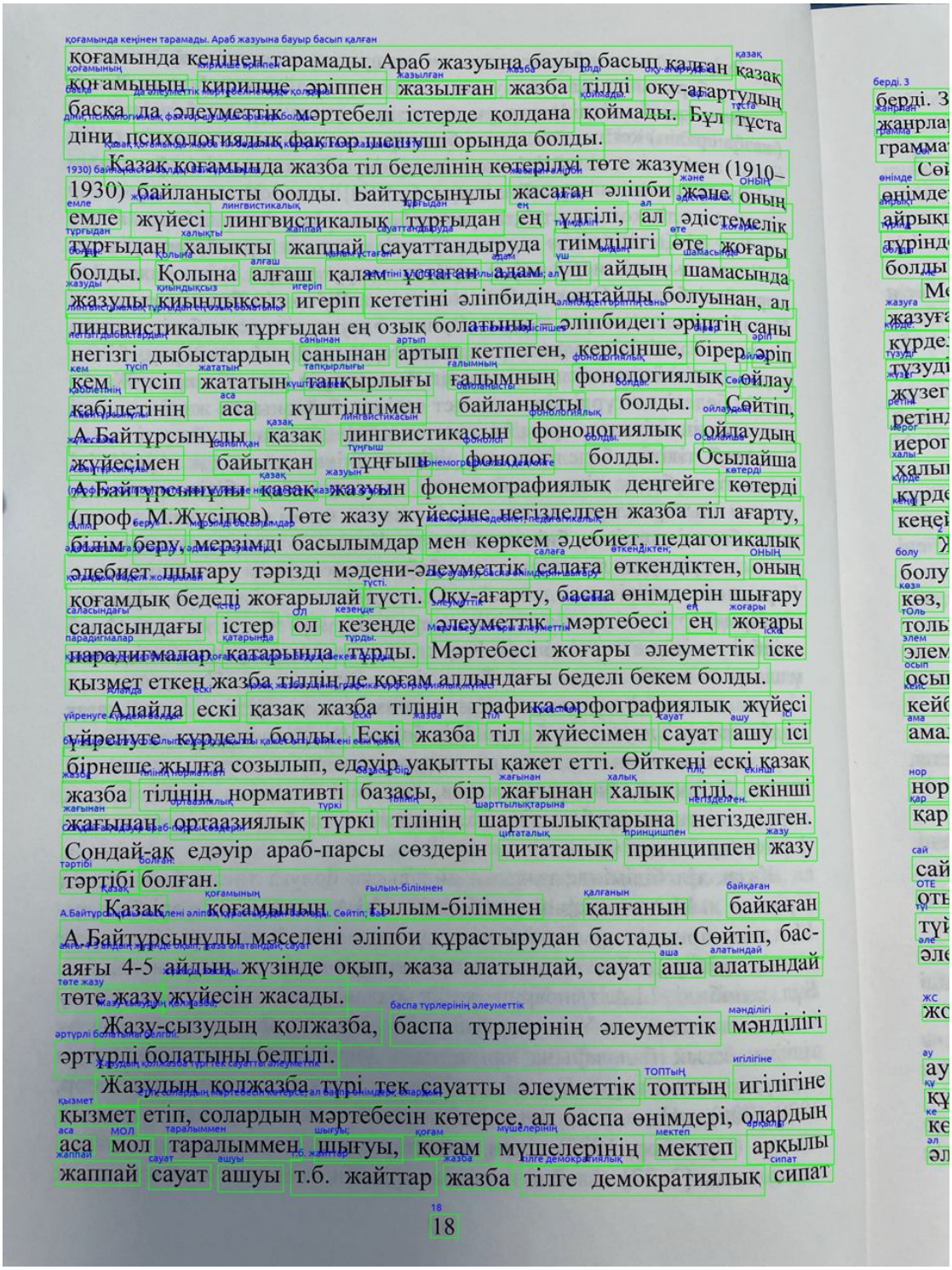

Figure 5.2 - High dimensional book page

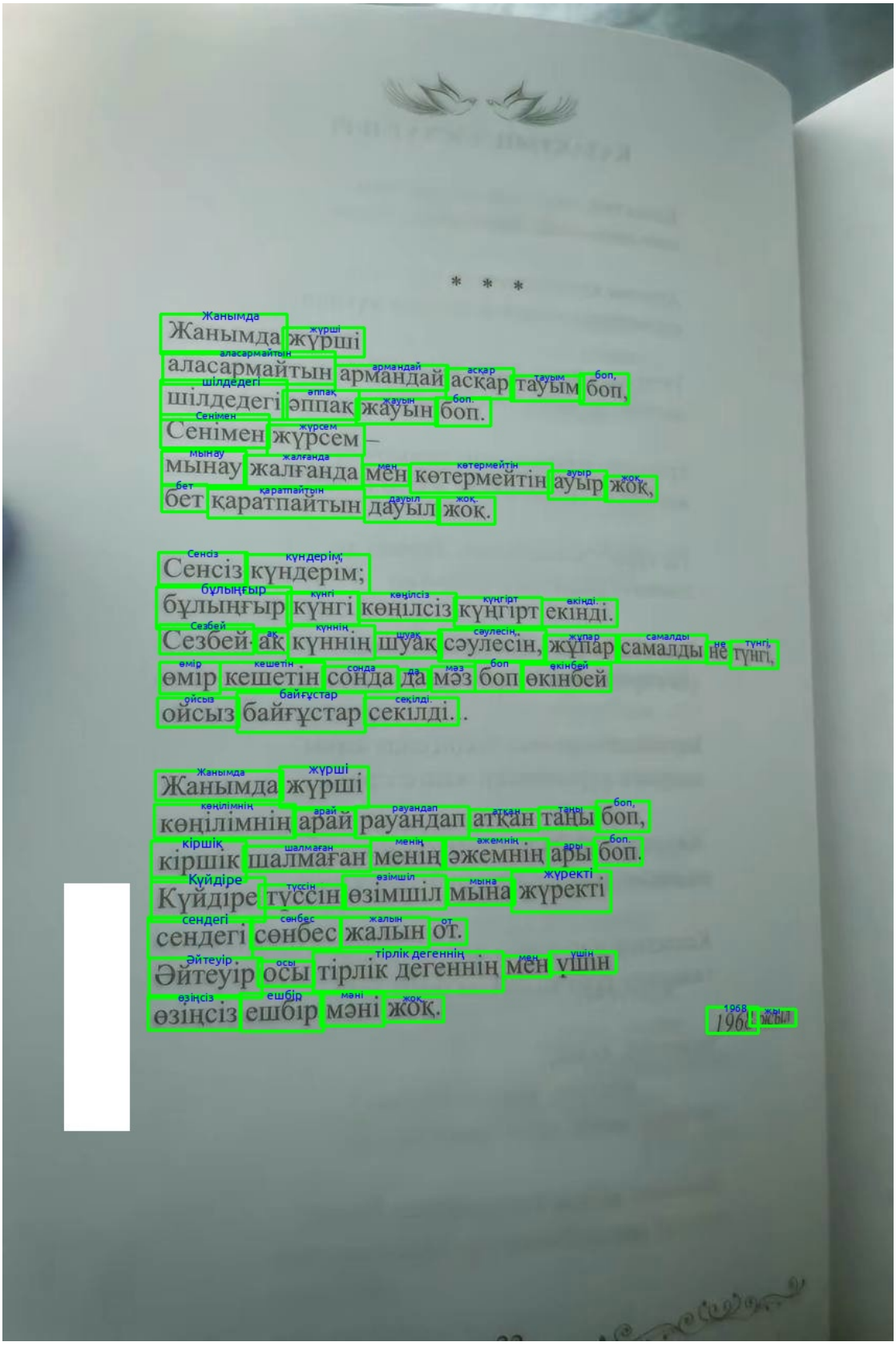

Figure 5.3 - Image with extra pictures and text

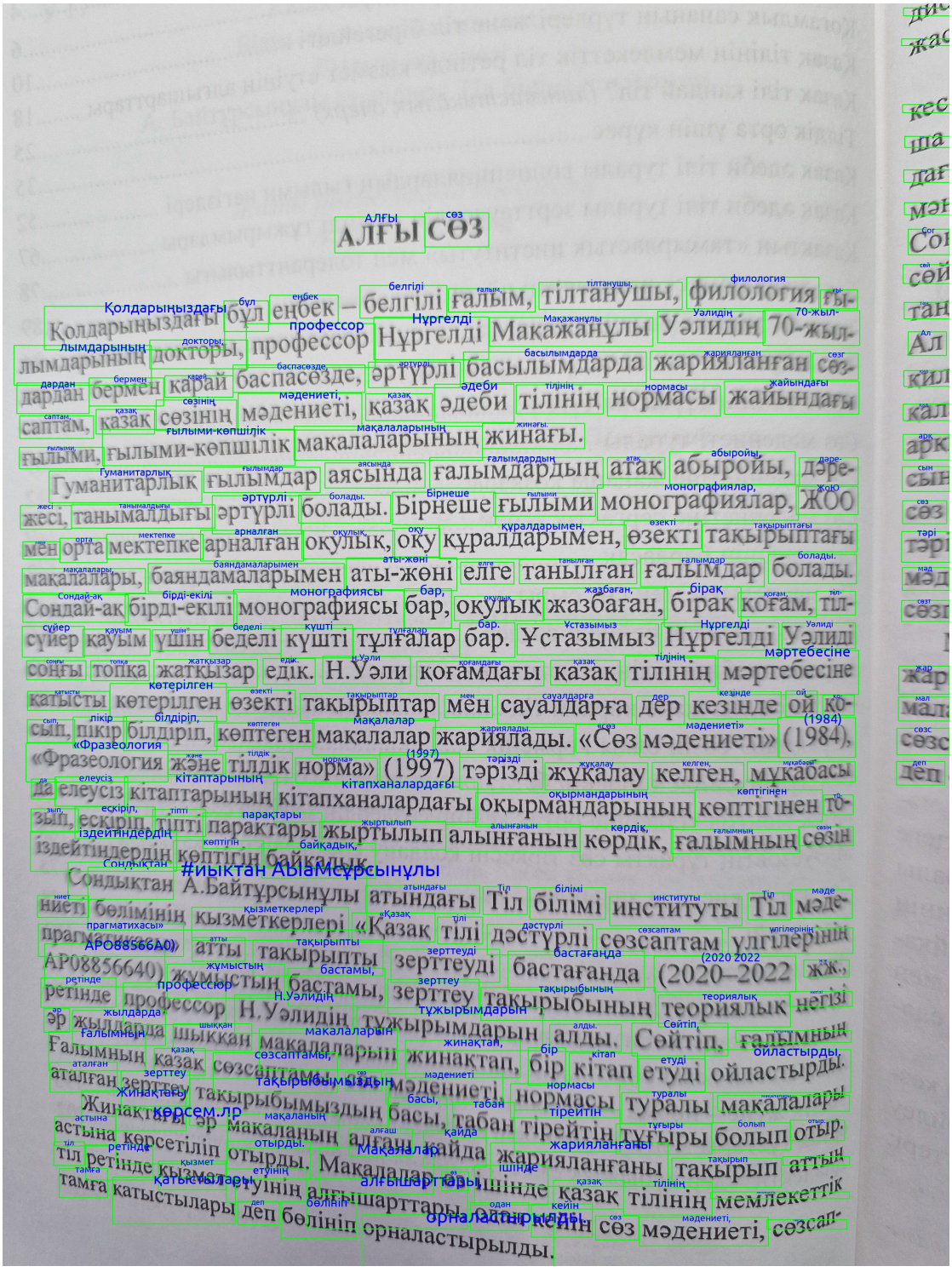

Figure 5.4 - Curved text, difficult conditions

Figure 5.5 - Book page with different colors and background

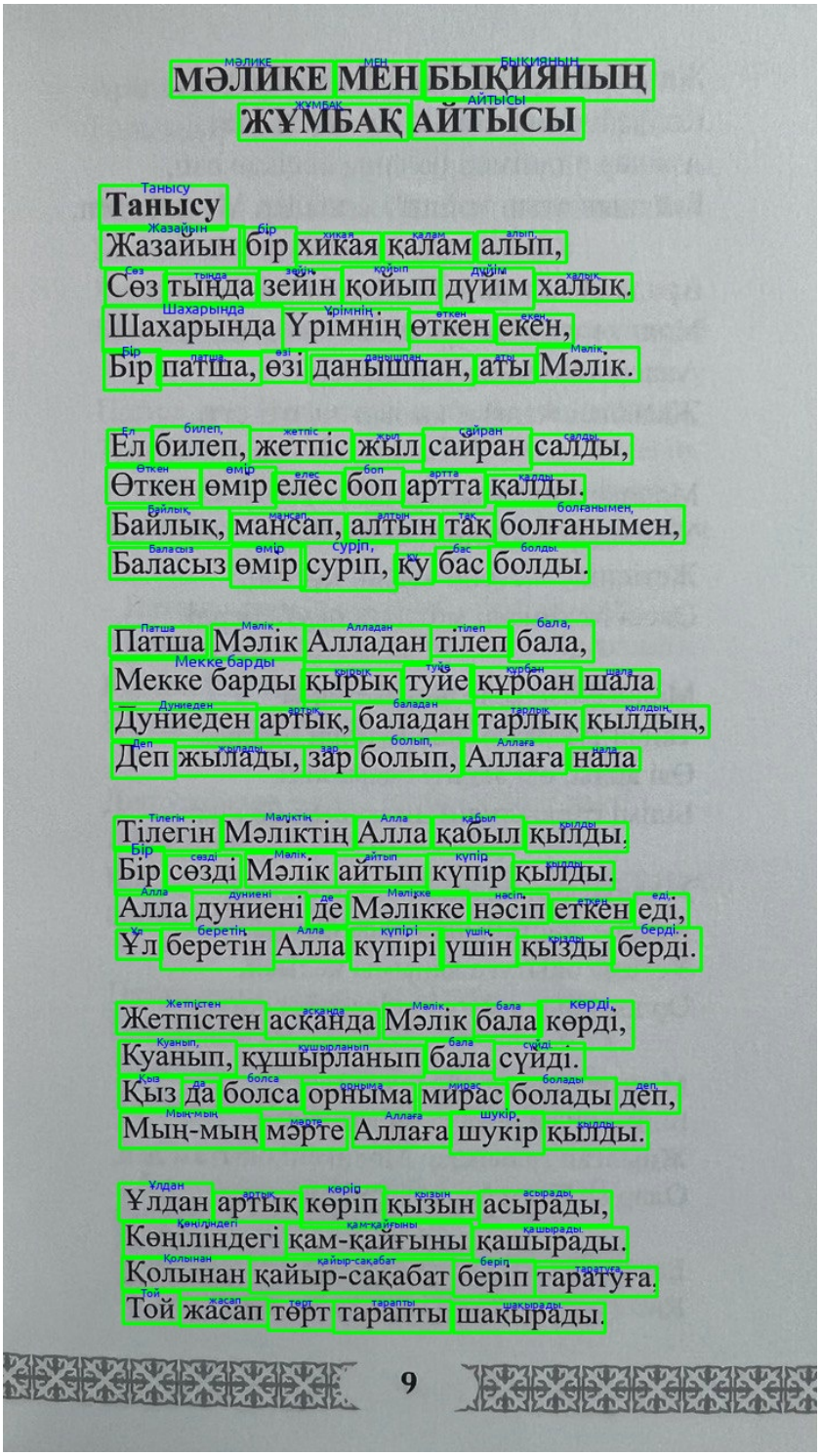

Figure 5.6 - Low dimensional book page

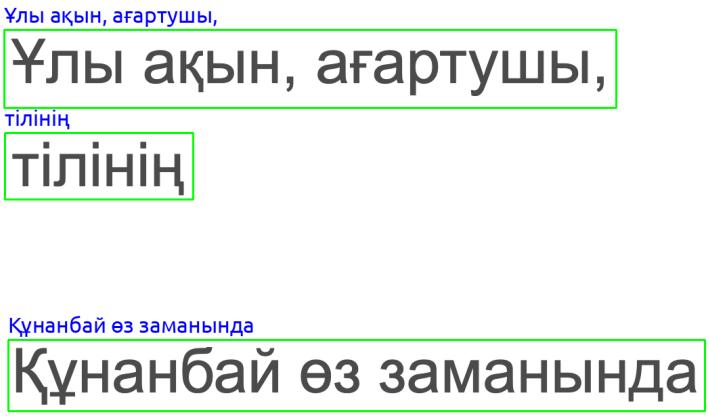

Figure 5.7 - Simple image with few words

```
image = Image.open(uploaded_file)
col1.image(image)
result = reader.readtext(np.array(image), paragraph=True)
...
```

This interactive platform not only serves as a testament to the OCR system's robustness but also provides a means for continuous improvement. User feedback and real-world usage data contribute to iterative enhancements of the OCR model, ensuring that it remains adaptable and effective against the evolving landscape of document formats and languages.

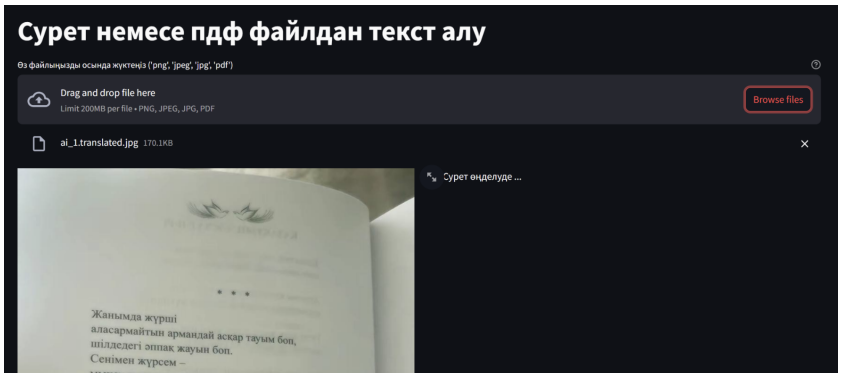

Figure 5.8 - Web application on huggingface

5.3 Discussion

The use of a synthetic dataset was pivotal in addressing the lack of real-world annotated texts for Kazakh, significantly enhancing the models' ability to recognize complex scripts. This success illustrates the potential of synthetic data in training OCR systems, particularly for languages that are underrepresented in digital

resources. Despite these advancements, the models still faced challenges with complex text layouts and stylized fonts found in literary works, highlighting areas for future improvement.

The potential applications of this OCR technology are vast, ranging from digital archiving and automated text processing to enhancing educational resources, thereby increasing access to Kazakh literary content. Future efforts will aim to refine these models by incorporating advanced deep learning techniques and expanding the training dataset to include a wider range of text variations.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis advanced the field of Optical Character Recognition (OCR) for the Kazakh language by developing a synthetic dataset and an OCR system tailored to its unique script characteristics. The study confirmed the effectiveness of the dataset in training robust models capable of high performance across various text environments, thereby enhancing the digitalization of Kazakh literary works and supporting digital archiving and education. Through meticulous dataset engineering and the integration of advanced deep learning techniques, the OCR models demonstrated high precision and efficiency in recognizing diverse text presentations, marking a significant step forward in preserving the cultural heritage of Kazakhstan. The research also highlighted the practical application of these models through the development of a web application, further illustrating their robustness and adaptability in real-world scenarios.

6.2 Future Work

Future research will focus on incorporating more sophisticated deep learning techniques and expanding the dataset to cover a broader range of text variations. Continuous improvement efforts, guided by user feedback and analysis of real-world usage, will aim to further enhance the system's adaptability to complex text layouts and fonts, ensuring the OCR technology remains at the forefront of digitization efforts for underrepresented languages. Additionally, future work should explore the integration of generative adversarial networks (GANs) to enhance the realism of synthetic datasets, thereby improving model performance in real-world applications.

Expanding the OCR system's capabilities to handle handwritten Kazakh texts and mixed-script documents, including the transition from Cyrillic to Latin scripts, is another vital direction. Moreover, implementing transfer learning techniques to leverage pre-trained models on larger multilingual datasets could further boost the OCR accuracy and efficiency.

The development of advanced post-processing techniques, such as error correction algorithms and contextual analysis, can significantly improve the quality of

the recognized text. Future research should also focus on enhancing the web application by integrating additional functionalities, such as automated translation and text-to-speech capabilities, to broaden its usability and accessibility.

Collaborative efforts with cultural and educational institutions to digitize and preserve historical documents will be essential in expanding the dataset and validating the OCR system's performance. Engaging with the Kazakh-speaking community for feedback and continuous improvement will ensure that the technology meets the needs of its users effectively.

In conclusion, while significant progress has been made, ongoing research and development efforts are crucial to fully realize the potential of OCR technology for the Kazakh language. By addressing the outlined future directions, the OCR system can become a vital tool in preserving and promoting the Kazakh language and its literary heritage in the digital age.

Bibliography

[1] Ray Smith. An overview of the tesseract ocr engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007. URL https://ieeexplore.ieee.org/document/4376991.

[2] Shunji Mori, Hirobumi Nishida, and Hiromitsu Yamada. Optical character recognition. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 15: 339–358, 2002.

[3] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014. URL https://arxiv.org/abs/1406.2227.

[4] Francisco Casacuberta, Luis Rodríguez, and David Llorens. Challenges of language diversity in the ocr of historical documents. *Digital Scholarship in the Humanities*, 36(2):243–256, 2021. doi: 10.1093/llc/fqaa024. URL https://academic.oup.com/dsh/article/36/2/243/5874251.

[5] Seiichi Uchida. Text recognition for languages with complex scripts. In *Document Analysis Systems (DAS)*, pages 15–25. Springer, 2019.

[6] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2315–2324. IEEE, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Gupta_Synthetic_Data_for_CVPR_2016_paper.html.

[7] Stephen V. Rice, George Nagy, and Thomas A. Nartker. *Optical Character Recognition: An Illustrated Guide to the Frontier*. Kluwer Academic Publishers, 1994.

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998.

[9] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20, 2016.

[10] Ian Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay

Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2014. URL https://arxiv.org/abs/1312.6082.

[11] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, 2008.

[12] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018.

[13] Xiangyu Zhu, Xiang Bai, and Cong Yao. Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science*, 10(1):19–36, 2016.

[14] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation and recognition. *arXiv preprint arXiv:1406.2227*, 2014. URL https://arxiv.org/abs/1406.2227.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014. URL https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[16] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019.

[17] Pengfei Wang, Chengquan Zhang, Fei Qi, Zuming Huang, Mengyi En, Junyu Han, Jingtuo Liu, Errui Ding, and Guangming Shi. A single-shot arbitrarily-shaped text detector based on context attended multi-task learning. *arXiv preprint arXiv:1908.05498*, 2019.

[18] 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2017.

[19] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. Rosetta: Large scale system for text detection and recognition in images. pages 71–79, 07 2018. doi: 10.1145/3219819.3219861.

[20] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12113–12122, 2020.

[21] Yongkun Du, Zhineng Chen, Caiyan Jia, Xiaoting Yin, Tianlun Zheng, Chenxia Li, Yuning Du, and Yu-Gang Jiang. Svtr: Scene text recognition with a single visual model. *arXiv preprint arXiv:2205.00159*, 2022.

[22] Rowel Atienza. Vision transformer for fast and efficient scene text recognition. In *International conference on document analysis and recognition*, pages 319–334. Springer, 2021.

[23] Shancheng Fang, Zhendong Mao, Hongtao Xie, Yuxin Wang, Chenggang Yan, and Yongdong Zhang. Abinet++: Autonomous, bidirectional and iterative language modeling for scene text spotting. *IEEE transactions on pattern analysis and machine intelligence*, 2022.

[24] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models. In *European conference on computer vision*, pages 178–196. Springer, 2022.

[25] Jonathan Baek et al. Easyocr: A practical and flexible ocr library. GitHub repository, 2020. URL https://github.com/JaidedAI/EasyOCR.

[26] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. *arXiv preprint arXiv:1904.01941*, 2019. URL https://arxiv.org/abs/1904.01941.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.

[29] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

[30] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45:2673–2681, 1997. URL https://api.semanticscholar.org/CorpusID:18375389.

[31] 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2016.