

## СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	3
РАЗДЕЛ 1. ВВЕДЕНИЕ .....	5
1.1 Пояснение темы.....	5
1.2 Цель работы .....	8
1.3 Актуальность работы .....	9
1.4 Задачи исследования.....	10
РАЗДЕЛ 2. ОСНОВНОЕ СОДЕРЖАНИЕ.....	11
2.2 Предлагаемый метод распознавания.....	11
2.2.1 Анализ литературы.....	11
2.2.2 Описание модели .....	11
2.2.3 Функция потерь.....	17
2.3 Подготовка данных .....	19
2.3.1 Анализ литературы.....	19
2.3.2 Подход к созданию синтетических данных .....	20
2.3.3 Используемые наборы данных .....	21
2.3.4 Методы наложения рукописей .....	22
2.3.5 Разделение данных .....	23
2.3.6 Результаты .....	24
2.4 Процесс обучения и результаты .....	25
2.4.1 Описание процесса .....	25
2.4.2 Параметры обучения .....	25
2.4.3 Аугментации данных.....	26
2.4.4. Валидационные метрики.....	26
2.4.5 Результаты .....	28
2.5 Проверка на реальных данных.....	31
2.5.1 Методология.....	31
2.5.2 Результаты и выводы.....	32
РАЗДЕЛ 3. ЗАКЛЮЧЕНИЕ.....	35

3.1 Выводы и ключевые результаты.....	35
3.2 Цели и задачи для дальнейшего исследования .....	36
СПИСОК ИСТОЧНИКОВ .....	38
ПРИЛОЖЕНИЯ .....	43

## Раздел 1. ВВЕДЕНИЕ

### 1.1 Пояснение темы

История распознавания текста восходит к разработке оптического распознавания символов в середине 20-го века. Технология OCR предполагала сканирование печатной страницы, определение каждой буквы и преобразование ее в машиночитаемый код. Ранние системы OCR были ограничены распознаванием только печатного текста, и скорость распознавания была низкой. С тех пор предложено огромное количество методов [Belaïd, Santosh, D'Andecy, 2013] для решения многих видов подзадач, таких как оптическое распознавание символов (англ. OCR), сегментация текстовых строк, анализ макета документа и так далее [Smith, 2007], [Ryu, Koo, Cho, 2014], [Koo, Cho, 2012].

Оцифровка рукописей и использование планшетов и ручек-стилусов привели к развитию технологий распознавания рукописного текста. Тем не менее, существуют области, в которых распознавание документов всё ещё вызывает трудности.


Например, распознавание заполненных от руки форм, рукописный текст на которых соседствует с печатным или даже залезает на него.

The image shows a medical form with a mix of printed and handwritten text. At the top, there is a header with contact information for the Arhangelskaya Oblast' Severovych. Below this, there is a section for patient information, including name, date of birth, and insurance details. The form is filled out with handwritten text in blue ink. There are several sections for medical information, including a section for the patient's medical history and a section for the doctor's notes. The form is signed by a doctor and has a blue circular stamp. The text is a mix of Russian and English, with some parts being handwritten and others printed.

Рисунок 1 — Медицинская форма с рукописным и печатным текстом

Многие исследователи решали эту проблему, распознавая сразу весь текст: для начала они отделяли рукописные (или напечатанные на машинке) тексты от фона [Li, Yin, Liu, 2018]. Исследователи извлекли связанные компоненты (англ. Connected Components — CCs) и векторы признаков для них, используя векторизованное распределение высот, ширин и расстояний между компонентами [Franke, Oberlander, 1993]. Наконец, они классифицировали каждый компонент, применяя классификатор k-ближайшего соседа (англ. k-NN). Тем не менее, большинство из этих традиционных методов использовали бинаризацию и извлечение связанных компонентов в качестве основных этапов предварительной обработки. Эти двухэтапные подходы имеют преимущества в том, что они позволяют нам использовать множество обычных алгоритмов компьютерной обработки изображений, но методы извлечения связанных компонентов подвержены ошибкам, когда два разных типа текстов накладываются друг на друга. Намного лучше в таких задачах сейчас себя показывают сверточные нейронные (англ. Convolutional Neural Networks — CNN) — недавно такие модели превзошли традиционные методы во многих областях [Long, Shelhamer, Darrell, 2015], [He и др., 2016], [Lin и др., 2017]. Например, группа исследователей использовала модели типа CNN для классификации связанных компонентов [Li, Yin, Liu, 2018].

Альтернативным подходом является разделение изображений рукописный и печатный текст, их раздельное распознавание и объединение полученного результата. Для этого формы размечают так, чтобы области с рукописным текстом были ограничены некими рамками, но это накладывает серьезные ограничения на подготовку формы и требует от заполняющего следования правилам. Таким образом, данный метод не применим к архивным документам и требует дополнительных вложений для организаций, собирающих какие-либо формы.



**Report of Medical Examination and Vaccination Record**  
 Department of Homeland Security  
 U.S. Citizenship and Immigration Services

**USCIS**  
**Form I-693**  
 OMB No. 1615-0033  
 Expires 07/31/2022

---

► **START HERE - Type or print in black ink.**

**Part 1. Information About You** (To be completed by the person requesting a medical examination, **NOT** the civil surgeon)

1. **Your Full Name**

Family Name (Last Name)

Given Name (First Name)

Middle Name
2. **Physical Address**

Street Number and Name

Apt. Ste. Flr. Number

City or Town

State

ZIP Code

(USPS ZIP Code Lookup)
3. **Other Information**

A. Gender  
☐ Male    ☐ Female

B. Date of Birth (mm/dd/yyyy)

C. City/Town/Village of Birth

Рисунок 2 — Медицинская форма с рамками для рукописного текста

Однако, разделять изображения на печатный и рукописный текст можно и с помощью компьютерной обработки. Как раз выделению из изображения рукописного текст будет посвящена данная работа.

При работе с оцифрованными изображениями для выделения рукописного текста возникает задача **сегментации рукописного текста**. Сегментация рукописного текста — это задача из области сегментации изображений. Если сегментация изображения — это сопоставление каждого пикселя изображения какому-то классу, то сегментация рукописного текста — это совокупность задач бинарной классификации для каждого пикселя: определение, является пиксель рукописным текстом или нет.

## 1.2 Цель работы

Целью данной работы является разработка интеллектуальной системы для сегментации рукописного текста с использованием передовых алгоритмов машинного обучения, таких как сверточные нейронные сети, для улучшения распознавания и обработки рукописных документов. Особое внимание уделяется алгоритму подготовки данных для машинного обучения и созданию системы, которая позволит готовить синтетические данные в области сегментации и распознавания рукописного текста.

NOTE TO ALL APPLICANTS: If you do not completely fill out this application or fail to submit required documents listed in the instructions, USCIS may deny your application.

art 14. Interpreter's Contact Information, Certification, and Signature

Provide the following information about the interpreter.

Interpreter's Full Name

Interpreter's Family Name (Last Name) Interpreter's Given Name (First Name)

Karim natural being Ishammud valid

Interpreter's Business or Organization Name (if any)

VIA. Vitamin power co

Interpreter's Mailing Address

Street Number Street Name City or Town State Zip

Mississippi Av, Green Building United states of America. Toronto Washington 650

Province DEI-MAC Postal Code Country Russia to India a

deviation. Per-pixel displacements are then computed using bicubic interpolation. Drop-out layers at the end of the contracting path perform further implicit data augmentation.

4) Experiments

We demonstrate the application of the u-net to three different segmentation tasks. The first task is the segmentation of neuronal structures in electron microscopic recordings. An example of the data set and our obtained segmentation is displayed in Figure 2. We provide the full result as Supplementary Material. The data set is provided by the EM segmentation challenge [14] that was started at ISBI 2012 and is still open for new contributions. The training data is a set of 30 images (512x512 pixels) from serial section transmission electron microscopy of the Drosophila first-instar larva ventral nerve cord (VNC). Each image comes with a corresponding fully annotated ground truth segmentation map for cells (white) and membranes (black). The test set is publicly available, but its segmentation maps are kept secret. An evaluation can be obtained by sending the predicted membrane probability map to the organizers. The evaluation is done by thresholding the map at 10 different levels and computation of the "warping error", the "Rand error" and the "pixel error" [14].

The u-net (averaged over 7 rotated versions of the input data) achieves without any further pre- or postprocessing a warping error of 0.0003529 (the new best score, see Table 1) and a rand-error of 0.0382.

This is significantly better than the sliding-window convolutional network result by Cireşan et al. [1], whose best submission had a warping error of 0.000420 and a rand error of 0.0504. In terms of rand error the only better performing

Рисунок 4 — Примеры страниц, на которых рукописный текст соседствует с печатным

NOTE TO ALL APPLICANTS: If you do not completely fill out this application or fail to submit required documents listed in the instructions, USCIS may deny your application.

art 14. Interpreter's Contact Information, Certification, and Signature

Provide the following information about the interpreter.

Interpreter's Full Name

Interpreter's Family Name (Last Name) Interpreter's Given Name (First Name)

Karim natural being Ishammud valid

Interpreter's Business or Organization Name (if any)

VIA. Vitamin power co

Interpreter's Mailing Address

Street Number Street Name City or Town State Zip

Mississippi Av, Green Building United states of America. Toronto Washington 650

Province DEI-MAC Postal Code Country Russia to India a

deviation. Per-pixel displacements are then computed using bicubic interpolation. Drop-out layers at the end of the contracting path perform further implicit data augmentation.

4) Experiments

We demonstrate the application of the u-net to three different segmentation tasks. The first task is the segmentation of neuronal structures in electron microscopic recordings. An example of the data set and our obtained segmentation is displayed in Figure 2. We provide the full result as Supplementary Material. The data set is provided by the EM segmentation challenge [14] that was started at ISBI 2012 and is still open for new contributions. The training data is a set of 30 images (512x512 pixels) from serial section transmission electron microscopy of the Drosophila first-instar larva ventral nerve cord (VNC). Each image comes with a corresponding fully annotated ground truth segmentation map for cells (white) and membranes (black). The test set is publicly available, but its segmentation maps are kept secret. An evaluation can be obtained by sending the predicted membrane probability map to the organizers. The evaluation is done by thresholding the map at 10 different levels and computation of the "warping error", the "Rand error" and the "pixel error" [14].

The u-net (averaged over 7 rotated versions of the input data) achieves without any further pre- or postprocessing a warping error of 0.0003529 (the new best score, see Table 1) and a rand-error of 0.0382.

This is significantly better than the sliding-window convolutional network result by Cireşan et al. [1], whose best submission had a warping error of 0.000420 and a rand error of 0.0504. In terms of rand error the only better performing

Рисунок 3 — Примеры страниц, на которых рукописный текст соседствует с печатным, весь печатный текст выделен красным (результат сегментации)

### **1.3 Актуальность работы**

Тема «Сегментации рукописного текста» не освещена в достаточной мере в существующих публикациях. Существует один пример подобной работы [Jo и др., 2020], но она не ссылается ни на какой программный код, с помощью которого можно было бы проверить полученные в работе результаты. Кроме того, в той работе не были продемонстрированы результаты работы системы на реальных данных.

В данной же работе, кроме создания систем подготовки данных и сегментации рукописного текста на изображениях, будет представлен исходный программный код и примеры работы системы на реальных примерах.

## **1.4 Задачи исследования**

Для достижения цели работы были выполнены следующие **задачи**:

1. Изучение литературы, касающейся темы исследования.
2. Изучение архитектуры нейронных сетей для сегментации изображений и функций потерь, выбор и реализация подходящих вариантов.
3. Определение алгоритма подготовки данных, реализация системы подготовки данных (фреймворка) и подготовка данных для обучения.
4. Обучение моделей (алгоритмов машинного обучения) и измерение качества.
5. Тестовое применение модели на реальных примерах и анализ результатов.



## **Раздел 2. ОСНОВНОЕ СОДЕРЖАНИЕ**

### **2.2 Предлагаемый метод распознавания**

#### **2.2.1 Анализ литературы**

Сегментация рукописного текста — сложная задача в области обработки изображений и анализа документов.

Всесторонний анализ литературы показывает, что сверточные нейронные сети (англ. CNN) стоит использовать для решения этой задачи благодаря их превосходной точности, устойчивости и обобщаемости.

Во-первых, статья [Afzal и др., 2015] продемонстрировала эффективность CNN для сегментации текста на уровне слов в рукописных документах. Они использовали глубокую архитектуру CNN, которая достигла самых современных результатов на базе данных IAM Handwriting Database — наборе данных, используемом для сравнения систем распознавания рукописного текста [Marti, Bunke, 2002]. Авторы пришли к выводу, что CNN могут точно распознавать рукописный текст даже в зашумленных и искусственно ухудшенных фотографиях документов [Renton и др., 2017].

Во-вторых, в научной статье [Renton и др., 2017] году, была предложена полностью конволюционная нейронная сеть для сегментации строк рукописного текста. Предложенная сеть, основанная на архитектуре U-Net [Ronneberger, Fischer, Brox, 2015], достигла конкурентоспособных результатов на общедоступных наборах данных рукописного текста, таких как IAM Handwriting Database [Marti, Bunke, 2002]. Авторы пришли к выводу, что CNN могут точно сегментировать текстовые строки в различных стилях почерка и с различными уровнями деградации.

В целом, анализ литературы показывает, что CNN весьма эффективны для задач, связанных с сегментацией изображений с рукописным текстом.

#### **2.2.2 Описание модели**

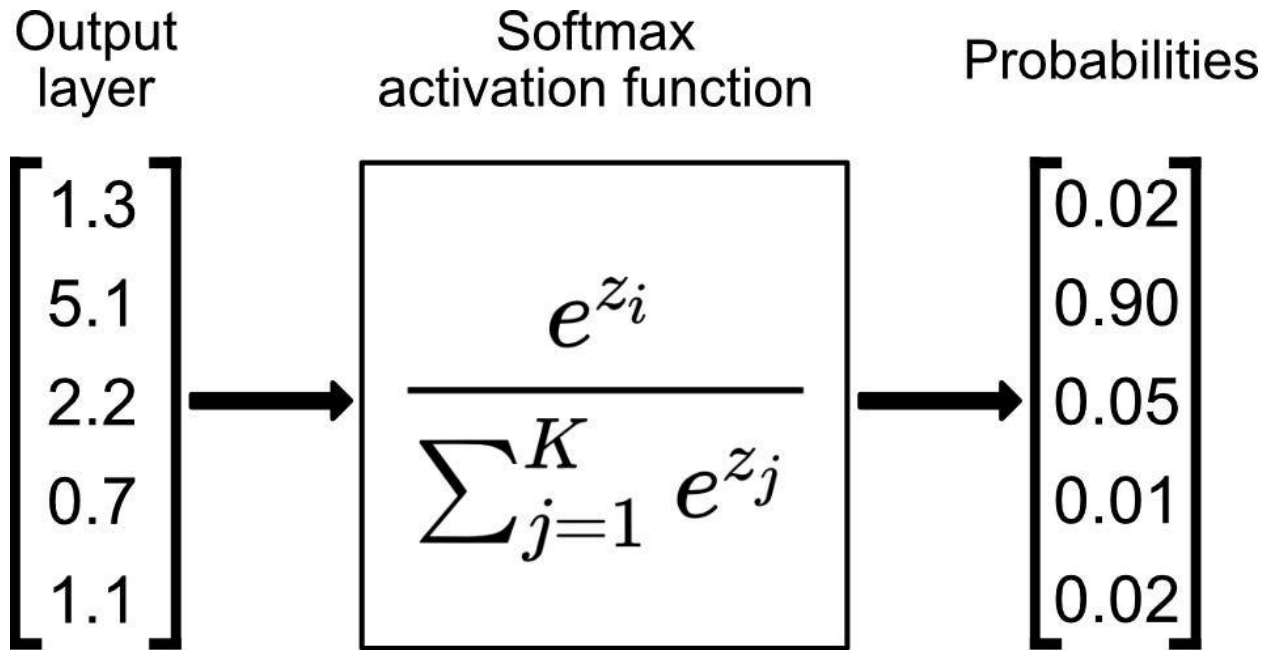
Архитектура U-Net состоит из сокращающегося пути, нижнего уровня (англ. bottleneck) и расширяющегося пути. Сокращающийся и

расширяющийся пути, в свою очередь, состоят из нескольких уровней, где каждый уровень состоит из двух частей: операции понижающей выборки (down-sampling) и набора сверточных слоев (англ. CNN layers). Целью операции понижающей выборки является уменьшение пространственного разрешения изображения и захват только наиболее заметных объектов. Далее, существует несколько уровней декодирования, которые состоят из набора сверточных уровней и операции повышения дискретизации. Операция увеличения выборки увеличивает пространственное разрешение карт объектов и объединяет их с объектами высокого разрешения из сокращающейся траектории. Это помогает восстановить пространственную информацию, которая была потеряна во время операции понижающей выборки сокращающегося пути.

Модель U-Net также включает в себя пропускные соединения (англ. skip-connections), которые соединяют соответствующие слои от сужающегося и расширяющегося путей. Эти соединения помогают сохранить пространственную информацию на картах объектов и повышают точность сегментации.

Конечный слой модели должен иметь те же пространственные размеры, что и входное изображение, и быть пропущен через функцию активации, а именно через многопеременную логистическую функцию (англ. SoftMax), также известную как SoftArgMax [Courville, 2016] и нормализованная экспоненциальная функция [Bishop, 2006], которая выдает оценку вероятности для каждого пикселя изображения принадлежать к определенному классу. Функцию иллюстрирует Рисунок

5.



В работе была использована модель типа U-Net, которая использовалась для end-to-end сегментации. Получая на вход 1-канальное изображение в градациях серого, модель отдавала на выход 1-канальную маску рукописного текста (насыщенность пикселя соответствовала вероятности пикселя принадлежать классу рукописного текста) такого же разрешения.

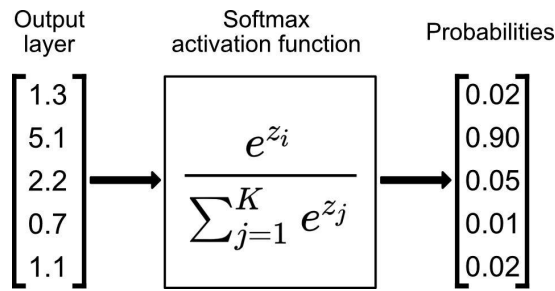


Рисунок 5 — Принцип работы функции SoftMax из [Radečić, 2022]

Каждый понижающий уровень в модели состоял из двух сверток с ядром 3x3, шагом (англ. stride) в 1 пиксель и отступом (англ. padding) в 1 пиксель и операции понижающей выборки; после каждого из сверточных слоев дополнительно стоял слой Batch normalization [Ioffe, Szegedy, 2015] и функция активации ReLU [Brownlee, 2019], [Liu, 2017]. Код представлен в Программный код 1.

```
class DoubleConv(nn.Module):
    def __init__(self, in_channels, out_channels) -> None:
        super(DoubleConv, self).__init__()

        self.conv = nn.Sequential(
            nn.Conv2d(in_channels, out_channels, kernel_size=3, stride=1, padding=1, bias=False),
            nn.BatchNorm2d(out_channels),
            nn.ReLU(inplace=True),

            nn.Conv2d(out_channels, out_channels, kernel_size=3, stride=1, padding=1, bias=False),
            nn.BatchNorm2d(out_channels),
            nn.ReLU(inplace=True),
        )
```

Программный код 1 — Реализация двух сверточных слоев с дополнительными слоями Batch Norm и ReLU

В качестве операции понижающей выборки использовалась операция выбора максимального пикселя для ядра размером 2x2 пикселя (англ. Max Pooling 2D), которая уменьшала на каждом шаге пространственное разрешение в 2 раза.

В качестве срединного слоя использовался слой «DoubleConv».

На расширяющемся пути использовались такие же уровни с двумя свертками и дополнительными слоями после них, но в качестве повышающей выборки использовался слой деконволюции (ConvTranspose2d) [Zeiler и др., 2010].

```
# down part
for feature in features:
    self.downs.append(DoubleConv(in_channels, feature))
    in_channels = feature

# bottleneck layer
self.bottleneck = DoubleConv(features[-1], features[-1] * 2)

# up part
for feature in reversed(features):
    self.ups.append(nn.ConvTranspose2d(feature * 2, feature, kernel_size=2, stride=2))
    self.ups.append(DoubleConv(feature * 2, feature))
```

Программный код 2 — Последовательность слоев модели

Первый уровень сокращающегося пути имел наименьшую глубину в свертках, а каждый следующий за ним — вдвое большую. Например, если всего в модели 4 уровня, то первый мог бы иметь глубину 16, второй — 32, третий — 64, четвертый — 128, а нижний уровень — 256. Расширяющиеся уровни в данном случае имели глубину 128, 64 и 32, соответственно, а последний слой (final layer) — 16. Skip-connections были реализованы с помощью конкатенации информации, полученной в момент сокращающегося пути (до применения операции понижающей выборки) и информации, поступающей на вход уровня расширяющегося пути. Программный код последовательности выполнения алгоритма «прямого прохода» представлен на Программный код 3. Полную же схему модели такого типа можно увидеть на Рисунок 6.

```

def forward(self, x):
    skip_connections = []

    # down
    for down in self.downs:
        x = down(x)
        skip_connections.append(x)
        x = self.pool(x)

    x = self.bottleneck(x)
    skip_connections = skip_connections[:-1]

    # up with skip connections
    for idx in range(0, len(self.ups), 2):
        # UP by ConvTranspose and Double conv on each iteration
        x = self.ups[idx](x)
        skip_connection = skip_connections[idx // 2]

        if x.shape != skip_connection.shape:
            x = F.interpolate(x, skip_connection.shape[2:],
                              mode='nearest', align_corners=False)
            x = skip_connection + x
            x = self.ups[idx + 1](x)
    
```

Программный код 3 — «Прямой проход» модели

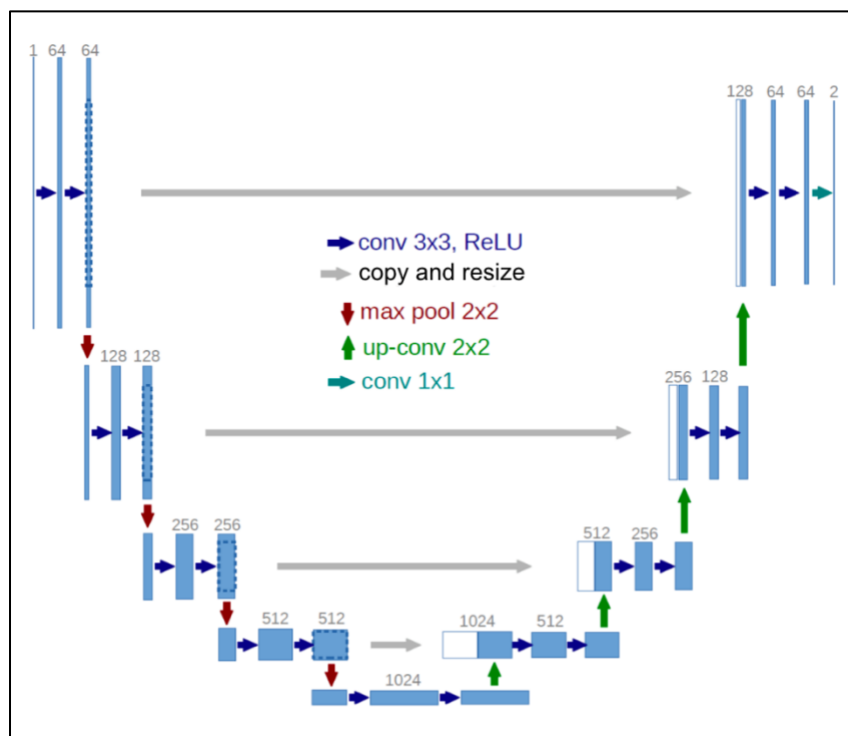


Рисунок 6 — Схема модели

### 2.2.3 Функция потерь

Функция потерь нужна для оценки того, насколько хорошо модель обучается на данных. Функция измеряет разницу между прогнозами модели и истинными значениями в обучающих данных. Цель заключается в достижении минимальной функции потерь.

В задачах сегментации изображений принято использовать функции потерь, которые оценивают сходство между маской классов, полученной от модели, и истинной маской сегментации на изображении.

Одной из наиболее распространенных функций потерь является перекрестная энтропия (cross-entropy) [Wikipedia contributors, 2023], которая работает для сегментации пиксельных масок.

Более сложные функции потерь, такие как Focal Loss [Lin и др., 2017] и Lovász-Softmax Loss [Berman, Triki, Blaschko, 2018] могут использоваться для решения проблем несбалансированности классов.

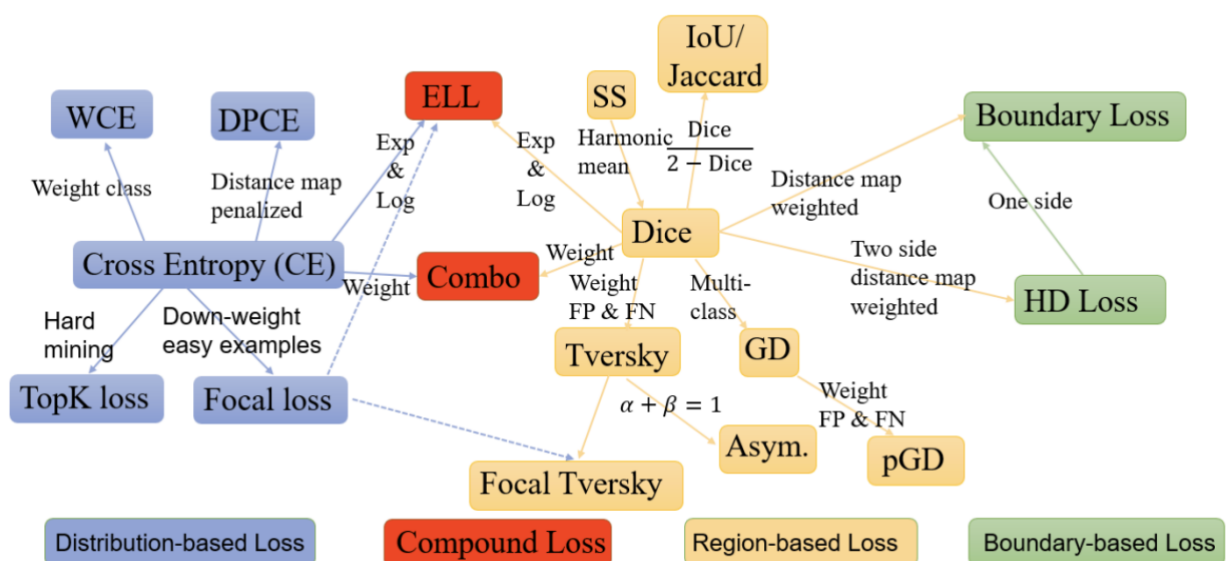


Рисунок 7 — Связь функция потерь [Ма и др., 2020]

Функция потерь Dice Loss может также использоваться для пиксельной классификации в наличии масок изображений класса. Dice Loss принимает во внимание размеры областей объектов и работает путем измерения сходства

между пиксельными областями. Dice Loss получается вычитанием из единицы коэффициента Дайса [Dice, 1945]

$$D_p = \frac{2 \sum_{i,j} p_{i,j} m_{i,j}}{\sum_{i,j} p_{i,j} + \sum_{i,j} m_{i,j}}$$

где

- $p_{i,j}$  — это предсказанная вероятность для пикселя принадлежать к целевому классу (от 0 до 1)
- $m_{i,j}$  — значение маски в пикселе (от 0 до 1)



## 2.3 Подготовка данных

### 2.3.1 Анализ литературы

Модели глубокого обучения, такие как CNN, требуют большого объема данных для обучения, поскольку у них есть много параметров, которые необходимо извлечь из данных. Эти параметры представляют характеристики и веса нейронной сети, которая может фиксировать сложные и нелинейные взаимосвязи в данных [Woodie, 2019].

Ученым часто приходится использовать синтетические данные для сегментации рукописного текста из-за отсутствия реальных данных, которые аннотируются на уровне пикселей [Dansena, Bag, Pal, 2021]. Аннотирование реальных данных для сегментации рукописного текста — утомительная и дорогостоящая задача, требующая от специалистов-людей пометить каждый пиксель изображения как принадлежащий рукописному или печатному тексту [Jo и др., 2020], [Souibgui и др., 2022]. Более того, реальные данные могут не охватывать все вариации и сценарии рукописного текста, такие как различные шрифты, языки, шрифты, стили, ориентация и фон [Dansena, Bag, Pal, 2021], [Li, Wu, Zhou, 2021].

Таким образом, использование синтетических данных — это хороший способ дополнить реальные данные и улучшения обучения глубоких нейронных сетей сегментации рукописного текста [Jo и др., 2020], [Davila и др., 2022], [Li, Wu, Zhou, 2021]. Синтетические данные могут быть сгенерированы с использованием существующих наборов рукописных символов или шрифтов и применением различных преобразований и эффектов для создания реалистичных изображений рукописного текста [Souibgui и др., 2022], [Davila и др., 2022], [Dansena, Bag, Pal, 2021].

Однако для данной задачи открытых данных нет вообще. Это также отмечают другие исследователи [Jo и др., 2020].

### 2.3.2 Подход к созданию синтетических данных

В данной работе был реализован метод создания синтетических данных, заключающийся в следующем:

1. Из страниц датасета с рукописными текстами создаются рукописи (фрагменты рукописного текста).

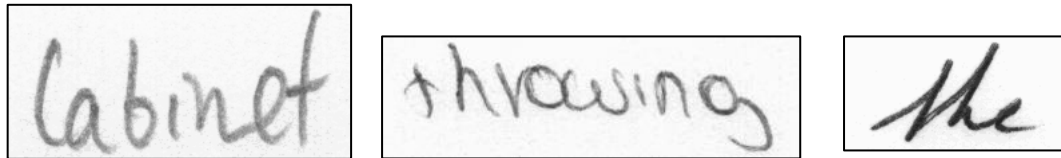


Рисунок 8 — Примеры фрагментов рукописей, полученные из IAM

2. На каждой странице печатного текста случайно размещается какое-то количество рукописей (фрагментов рукописного текста) с применением методов компьютерных преобразований изображений.

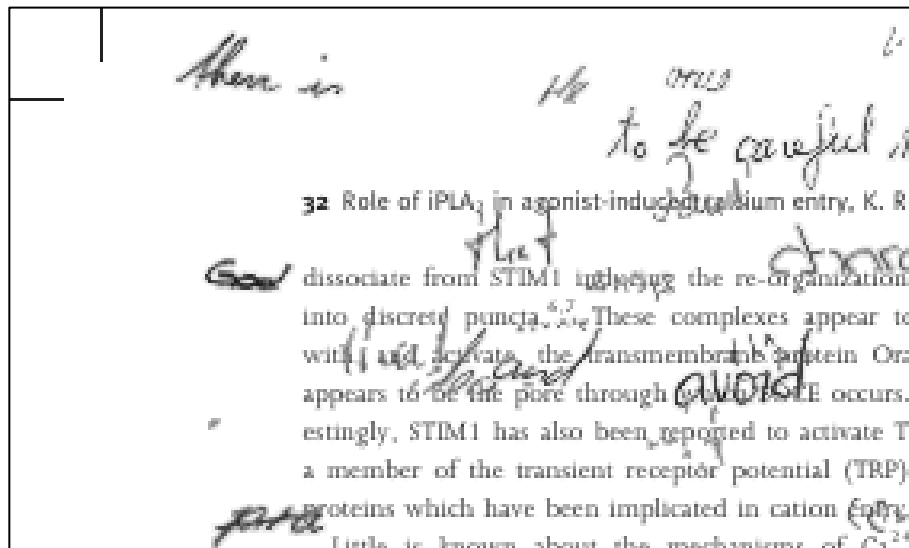


Рисунок 9 — Фрагмент печатной страницы с наложенными рукописями

3. Для каждой страницы готовится попиксельная маска рукописного текста.

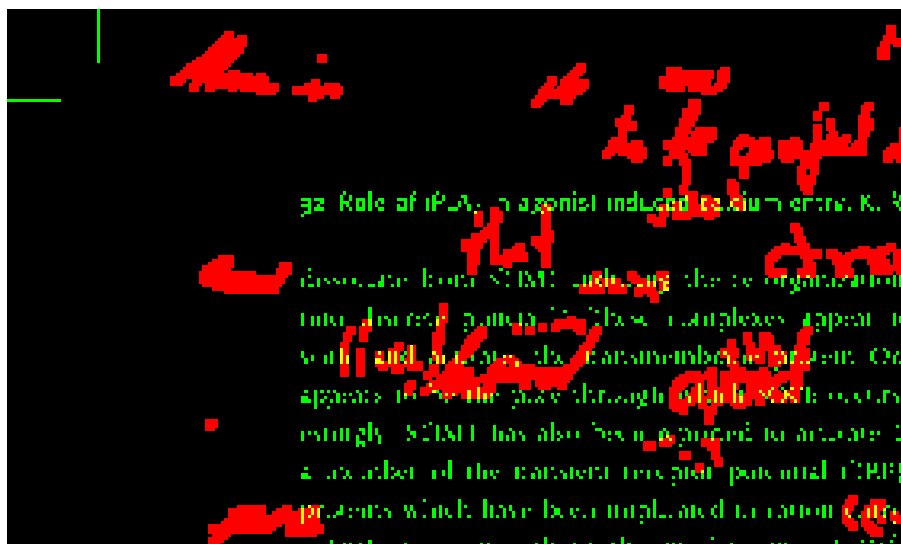


Рисунок 10 — Фрагмент маски для страницы рукописного текста  
(красный — пиксели рукописного текста)

Предполагается использовать существующие наборы (англ. dataset) данных с печатными и рукописными текстами, а синтезировать только смешанный dataset — dataset с рукописями, наложенными на печатные страницы.

### 2.3.3 Используемые наборы данных

Были использованы следующие dataset:

- IAM — Набор данных с рукописями с рукописями.
- PubLayNet — Набор данных с рукописями с отсканированными документами.

IAM dataset — это набор данных, используемый для задач распознавания рукописного ввода [Marti, Bunke, 2002]. Он содержит образцы рукописного текста в формате фотографий страниц А4. Набор данных включает тексты на различных европейских языках и предназначен для исследования и сравнения производительности различных методов распознавания текста.

В работе использовался вариант IAM с 2197 страниц. Пример страницы из IAM на в Приложениях (Рисунок 18).

PubLayNet — это открытый набор данных, содержащий данные, для аннотации и классификации многостраничных научных статей и других типов документов [Zhong, Tang, Jimeno Yepes, 2019]. Он содержит более 900 000 изображений и связанных с ними аннотаций, в основном в формате COCO. Набор данных включает в себя 24 класса документов и может быть использован для обучения модели машинного обучения в таких задачах, как автоматическое распознавание текста и классификация научных документов.

В работе использовалась часть этого набора данных объемом в 10081 страниц. Пример страницы в Приложениях (Рисунок 19).

### 2.3.4 Методы наложения рукописей

Для реалистичного синтеза данных основной задачей было сохранение текстур изображений рукописного текста и шумов оригинальных документов. Во-первых, следует отметить, что текстуры изображений рукописного текста могут быть решающим признаком, позволяющим отличить их от текстов, напечатанных машинным способом. Во-вторых, необходимо сохранить постоянный шум, унаследованный от процесса сканирования, чтобы уменьшить расхождения между распределениями синтетических и реальных данных. Если просто добавить изображение рукописного текста на изображение печатного текста, то фон будет более насыщенным, и большинство шумов сканирования исчезнут.

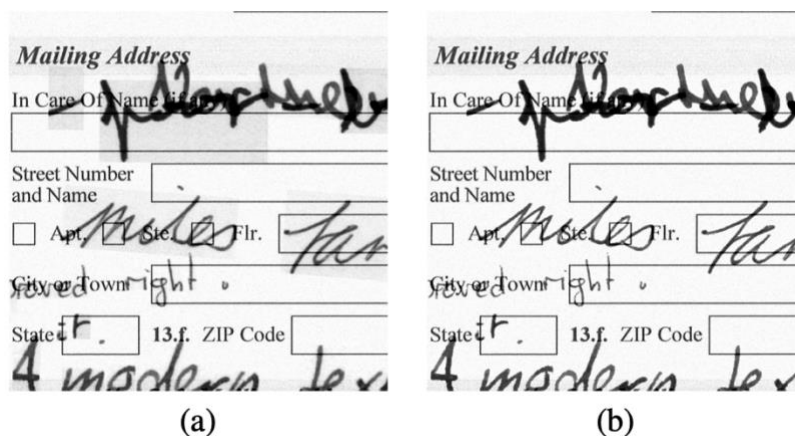


Рисунок 11 — Наложение рукописей на отсканированный документ

В статье [Jo и др., 2020] описывается подход, который позволяет решить эту проблему. Для каждой рукописи создается бинарная маска, которая потом используется для наложения рукописи на страницу печатного текста особым образом.

Непосредственно алгоритмов наложения есть несколько:

- Как описан в статье [Jo и др., 2020]

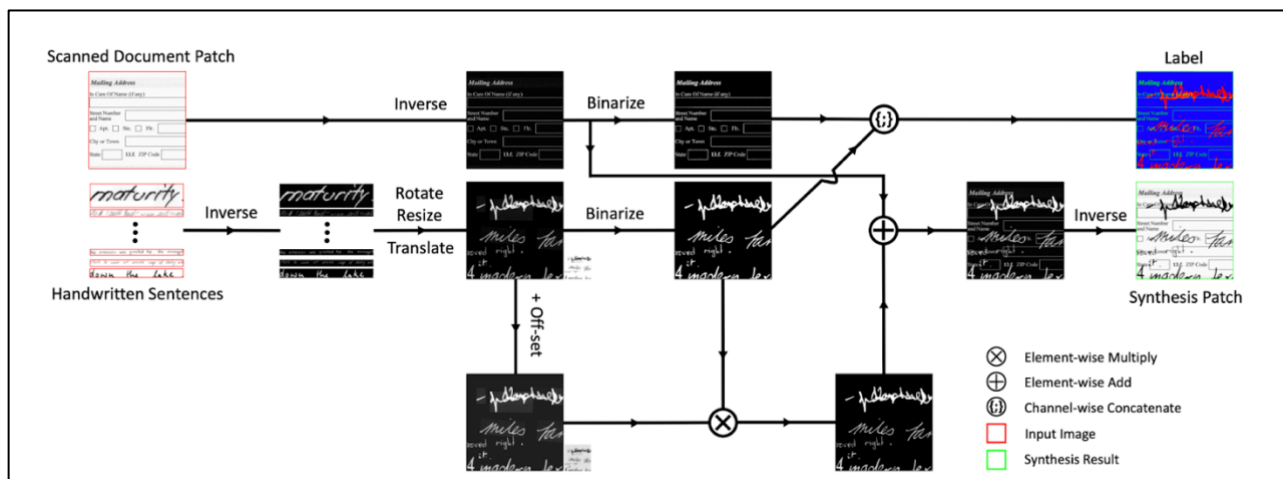


Рисунок 12 — Схема подготовки данных из статьи [Jo и др., 2020]

- Альфа-смешивание (англ. alpha-blending) [Porter, Duff, 1984]
- Алгоритм, реализованный в данной работе.

Алгоритм, реализованный в данной работе похож на алгоритм из статьи [Jo и др., 2020], однако при наложении инвертированных изображений фрагмента рукописи и страницы документа используется попиксельный максимум, а не сложение. Исходный код представлен в приложениях (Программный код 4).

### 2.3.5 Разделение данных

При машинном обучении крайне важно разделять данные так, чтобы выборки, используемые для обучения и для тестирования, не пересекались. Это нужно для предотвращения переобучения и повышений общности модели [Agrawal, 2021].

В данной работе данные разделялись на 3 выборки: обучающая (train), валидационная (val) и тестовая (test). Для этого исходные датасеты (печатный и рукописный) были разделены на 3 выборки, а при генерации каждой из выборок смешанного датасета использовались соответствующие выборки исходных датасетов. Таким образом рукописи со страниц, на которых модель тестировалась, не использовались в обучении.

### **2.3.6 Результаты**

Всего, из 2197 страниц рукописного текста было получено 47 538 фрагментов рукописей. Далее для каждой из 10081 страниц печатного текста случайным образом выбиралось 300 фрагментов рукописей (повторы разрешались в пределах выборки). Таким образом, один и тот же рукописный фрагмент использовался, в среднем, 63 раза. Размер полученного датасета составил 10081 страница с масками. Размеры выборок составили, соответственно:

- Обучающая: 7057 страниц (7/10 от общего объема),
- Валидационная: 1008 страниц (1/10 от общего объема),
- Тестовая: 2016 страниц (2/10 от общего объема).

Кроме того, результатом является создание фреймворка для генерации подобных смешанных датасетов. Фреймворк включает в себя:

- Метод автоматического получения фрагментов рукописей из страницы рукописного текста, использующий морфологические преобразования и другие методы компьютерных преобразований изображений. Код представлен в Приложениях (Программный код 5).
- Метод размещения рукописных фрагментов на странице печатного текста (Программный код 4).
- Функции и классы, позволяющие автоматизировать процесс создания подобных наборов данных до создания нескольких объектов и вызовов нескольких методов (Пример в файле «MIX\_dataset\_preparing.ipynb»).

## 2.4 Процесс обучения и результаты

### 2.4.1 Описание процесса

Обучение нейронной сети — это процесс, при котором нейронная сеть приспособляется к решению конкретной задачи, благодаря обработке большого количества данных.

В ходе обучения нейронной сети, её параметры (веса и смещения) настраиваются таким образом, чтобы минимизировать ошибку предсказания сети на обучающих данных. Обучающие данные представляют собой наборы входных и выходных данных, на которых сеть тренируется.

Процесс обучения состоит из нескольких эпох, а эпохи — из шагов. На каждом шаге сеть делает предсказание на обучающих данных, сравнивает его с истинным значением и корректирует свои параметры для уменьшения ошибки предсказания. Одна эпоха — это совокупность шагов, которые охватывают всю обучающую выборку. Если сеть обучалась 5 эпох, то это значит, что она 5 раз предсказала результаты по всей обучающей выборке.

При обучении важную роль имеют функция ошибки (англ. loss), оптимизатор, который меняет веса модели при обратном распространении ошибки (англ. back propagation) и выбранные исследователем валидационные метрики, которые измеряют «качество» модели.

### 2.4.2 Параметры обучения

Если говорить про функцию ошибки, то, хотя градиенты, рассчитываемые при обратном распространении ошибки, проще у перекрестной энтропии, эмпирически была выбрана функция потерь Dice Loss, потому что с её помощью удалось достичь результатов. Функция потерь была детально описана в разделе про модель (2.2.3 Функция потерь).

В качестве оптимизатора был выбран Adam [Kingma, Ba, 2014] с начальным шагом  $1e-2$ . Для оптимизации обучения был использован планировщик шага ReduceLROnPlateau [ReduceLROnPlateau, 2023].

Для автоматизации обучения использовались пакеты PyTorch [Paszke, Gross, Massa, 2019] и PyTorch Lightning [Lightning team, Falcon, 2019].

Картинки на вход подавались в разрешении 800x600. Обучение проходило пакетами (англ. batch) по 4–8 изображений.

Обучение производилось на сервере с использованием графического ускорителя (англ. GPU) Nvidia 2080ti, для визуализации процесса обучения использовался TensorBoard из пакета TensorFlow [TensorFlow Developers, 2023].

### **2.4.3 Аугментации данных**

Аугментация данных — это процесс создания новых примеров данных путем внесения небольших случайных изменений в существующие данные. Эта техника повышает разнообразие и количество обучающих примеров и может быть полезна для уменьшения переобучения и улучшения обобщающей способности модели.

В качестве аугментаций в данной работе использовались следующие преобразования:

- Случайный поворот в пределах 35 градусов,
- Отражение по горизонтали с вероятностью 0.2,
- Отражение по вертикали с вероятностью 0.2.

Эти аугментации были реализованы с помощью пакета Albumentations [Buslaev и др., 2020].

### **2.4.4. Валидационные метрики**

В качестве валидационных метрик в работе были использованы Accuracy, Precision, Recall, F1-score, которые выражаются через количество TP, FP, FN, TN (определения будут даны ниже) результатов тестов. Единичным результатом теста считается бинаризованное значение пикселя предсказания модели. Бинаризация осуществляется с порогом 0.5. Таким образом, для вычисления значения метрик и величин TP, FP, FN, TN используется сумма по всем пикселям по всем изображениям выборки.



- $TP$  — количество верно предсказанных результатов целевого класса (пикселей со значением 1, совпадающих по значению в предсказании и в маске);
- $TN$  — количество верно предсказанных результатов не целевого класса (пикселей со значением 0, совпадающих по значению в предсказании и в маске);
- $FP$  — количество негативных результатов (пикселей со значением 0, не принадлежащих к целевому классу), ошибочно отмеченных при тесте; ошибка первого рода, потому что верная нулевая гипотеза была отвергнута;
- $FN$  — количество позитивных результатов (пикселей со значением 1, принадлежащих к целевому классу), пропущенных при тесте; ошибка второго рода, потому что нулевая гипотеза была неверно принята;

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{2 * Recall * Precision}{Recall + Precision} = \frac{2 * TP}{2 * TP + FP + FN}$$

Интересно заметить, что если считать результаты предсказаний для всех пикселей бинарными, а не вероятностными: 0 или 1, то коэффициент Дайса соответствует в точности метрике точности теста, называемая F1-score с показателем степени 1 [Wikipedia contributors, 2023], которую часто используют для измерения качества модели.

$$D_{binary} = F_1 = \frac{2 * TP}{2 * TP + FN + FP}$$

Преимущество метрики F1-score заключается в том, что она учитывает и полноту (Recall), и точность модели (Precision). Стоит отметить, что целевая метрика в разных задачах может быть разной. В данной же работе F1-score была выбрана целевой.

#### 2.4.5 Результаты

Были обучены несколько моделей с архитектурой, описанной в разделе про модели (2.2.2 Описание модели). Примеры генерируемых моделью масок представлены ниже (Рисунок 13).

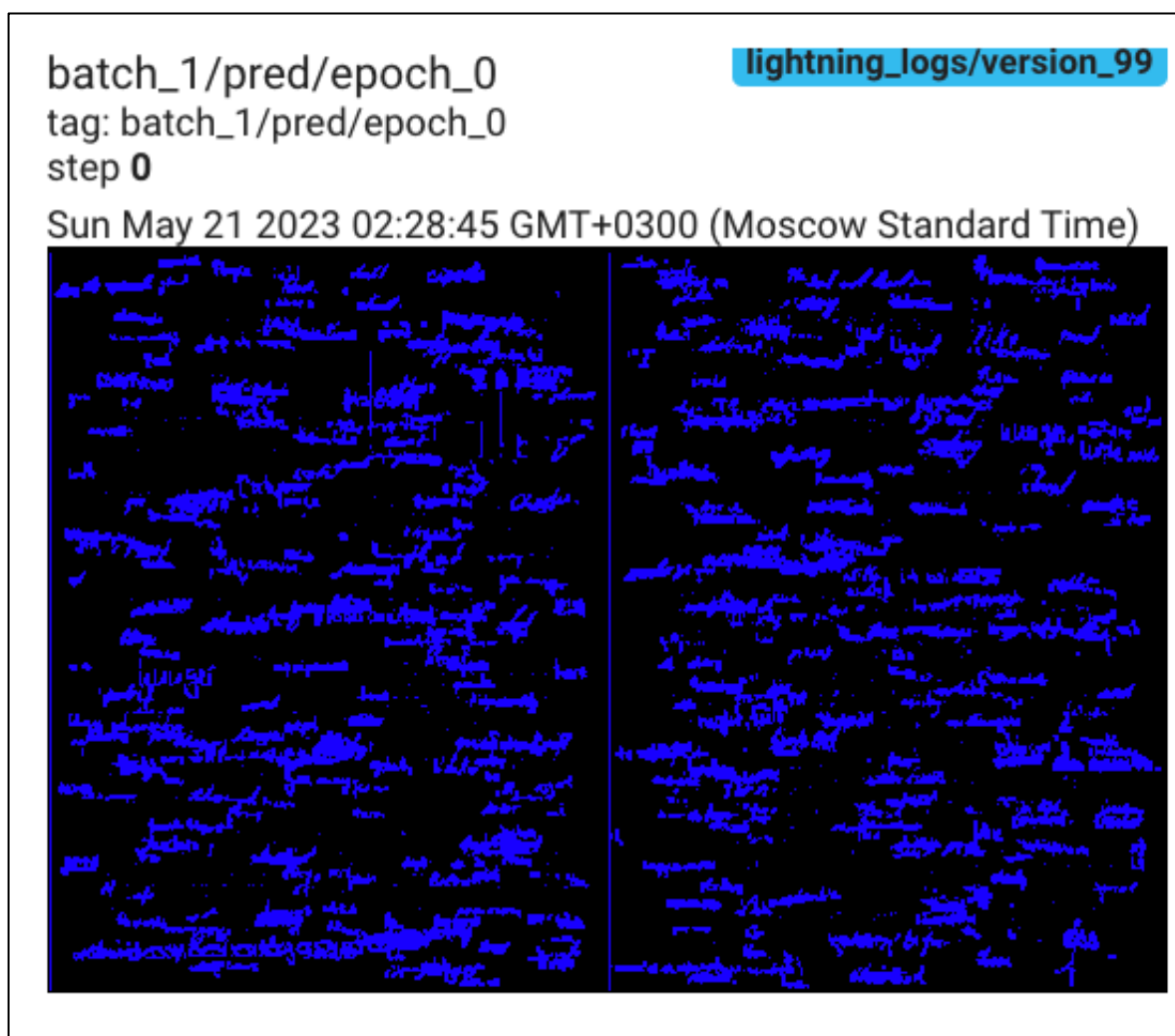


Рисунок 13 — Предсказанная моделью маска рукописного текста

Обучение каждой модели занимало несколько часов астрономического времени. За это время было пройдено 25 эпох. По графикам видно, что после 7 эпохи, качество предсказаний модели уже сильно не улучшалась.

Результаты обучения представлены ниже (Таблица 1). Цифры в названии моделей — это глубина уровней сокращающегося пути в модели.

Таблица 1 — Результаты обучения

Модель	Кол-во параметров	Accuracy	Precision	Recall	F1-score
Unet 16-32-64-128 (128)	1.9 М	0.976	0.942	0.892	0.916
Unet 32-64-128-256 (124)	7.8 М	0.979	0.949	0.910	0.929
<b>Unet 64-128-256-512 (123)</b>	<b>31.0 М</b>	<b>0.983</b>	<b>0.959</b>	<b>0.924</b>	<b>0.941</b>

Модели с текущими параметрами и подходами к обучению можно считать обученными, так как приближаясь к 25 эпохе значения метрик уже почти не меняются. Три лучших слепка каждой из модели были сохранены отдельно как чекпоинты (англ. checkpoints).

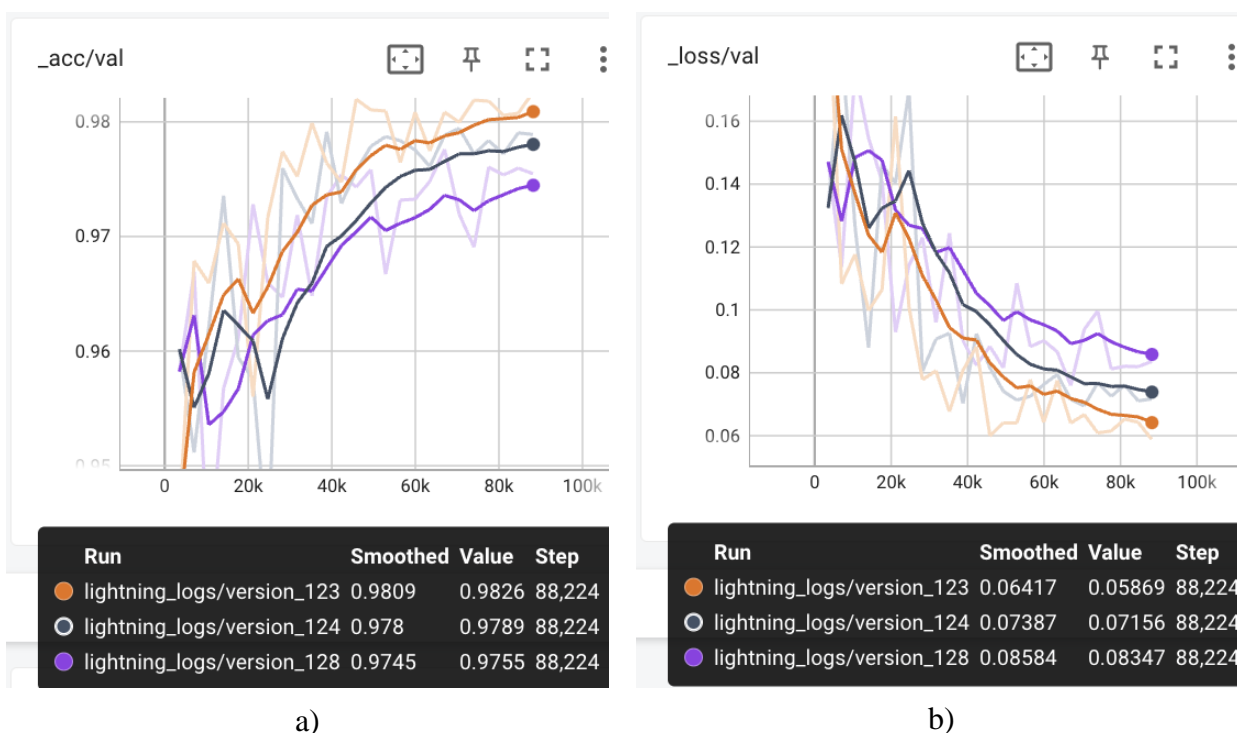


Рисунок 14 — Валидационные метрики при обучении (a — Accuracy, b — loss); на графике отражены 25 эпох (88 тыс. шагов), масштаб по горизонтали линейный

Лучшая модель по всем метрикам — это модель с наибольшим количеством параметров «Unet 64-128-256-512 (123)». Скорее всего, большее количество параметров позволяет ей лучше запоминать особенности данных.

Как видно, модель действительно очень хорошо сегментирует рукописный текст (ПРИЛОЖЕНИЯ, Рисунок 20). Невооруженным глазом даже не заметны участки, в которых модель ошибается.

## 2.5 Проверка на реальных данных

### 2.5.1 Методология

Для количественной проверки именно задачи сегментации рукописного текста нам нужны были бы размеченные наборы данных, которых нет, а готовить дорого (2.3.1 Анализ литературы). Альтернативными способами проверки является проверка в каких-то других задачах, в которых задача сегментации является промежуточной.

Например, есть гипотеза, что качество распознавания будет выше, если сначала сегментировать весь рукописный текст, вырезать его на отдельный слой, и только потом распознавать. Для проверки таким способом можно сравнить качество распознавания двумя способами:

- Сначала **детектировать** рукописи с помощью специальных алгоритмов (англ. Handwritten detector) а потом отмеченные участки (англ. bounding boxes) распознавать (англ. Handwritten recognizer);
- Сначала **сегментировать** рукописи с помощью специальных алгоритмов (англ. Handwritten detector) а потом распознавать по маске рукописного текста (англ. Handwritten recognizer)

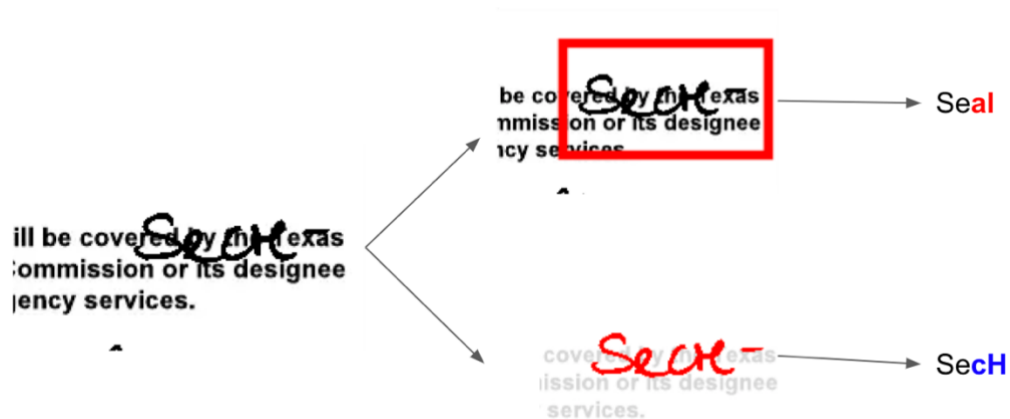


Рисунок 15 — Сравнение распознавания рукописного текста с использованием детекции и сегментации

В данной работе из-за своей простоты и наглядности было решено проверять работу сегментации качественно. Для этого были подготовлены

несколько заготовок, которые распознавались разными моделями (ПРИЛОЖЕНИЯ, Рисунок 21).

## 2.5.2 Результаты и выводы

Проверка на реальных данных позволяет сделать следующие выводы.

Во-первых, на отсканированных изображения модель почти ничего не распознает. Предполагаю, что это может быть связано со способом синтеза данных (при обучении использовались фотографии рукописей, а не сканы). Если сфотографировать документ, то модель работает лучше сегментирует рукописи. Если же к фотографии применить обработку, которая сделает её похожей на скан, например, используя приложение Adobe Scan, то сегментировано будет уже меньшее количество рукописного текста, чем на фотографии, но всё ещё лучше, чем на скане

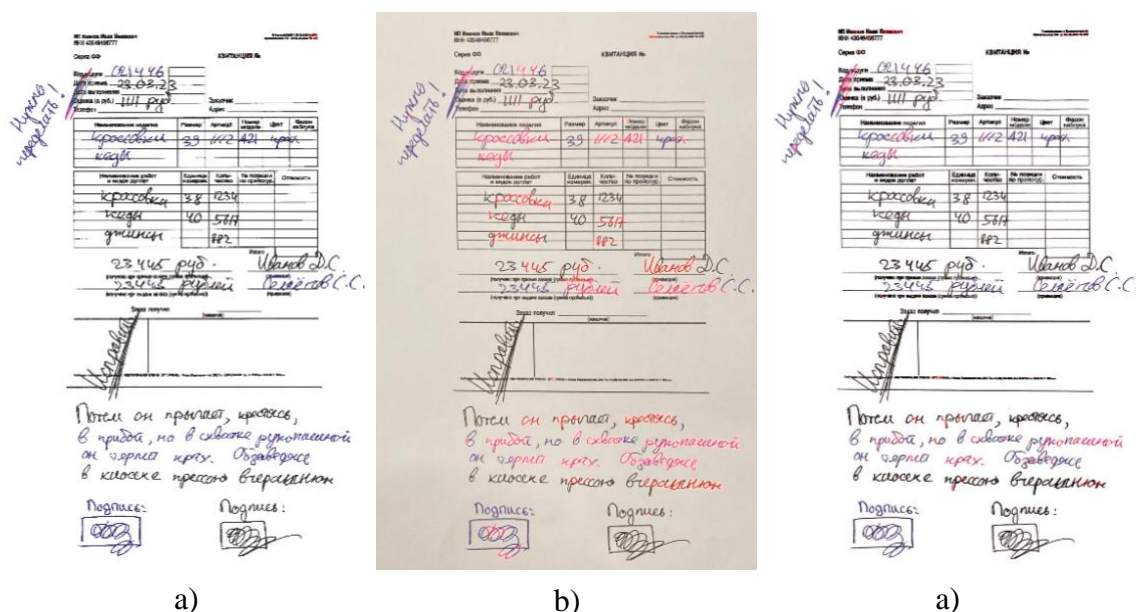


Рисунок 16 — Работа лучшей модели (128) на тестовом примере (из Рисунок 21а в ПРИЛОЖЕНИЯ); а — скан, б — фотографий, с — преобразованная фотография; рукописи выделены красным

Во-вторых, на фотографиях ни одна из моделей не распознает большинство рукописей, видно много пропусков и даже редкие ложные срабатывания (ПРИЛОЖЕНИЯ, Рисунок 22).

В-третьих, модели с меньшим количеством параметров лучше распознают рукописный текст на реальных данных. Скорее всего, модели с большим числом параметров сильнее переобучаются на специфичном тренировочном наборе данных.

Таблица 2 — Лучшие слепки моделей на реальных тестовых примерах (ПРИЛОЖЕНИЯ, Рисунок 21 — Заготовки для проверки (a, b, c, d)); здесь в модели указана одна и та же модель, но чекпоинты из разных эпох.

	Картинка a)	Картинка b)	Картинка c)
Номер лучшей модели по качественной проверке	128 (19 эпох)	128 (14 эпох)	256 (20 эпох)

В-четвертых, модели лучше сегментируют текст, написанный синими чернилами, несмотря преобразование в градации серого на входе в модель. Это можно объяснить тем, что яркость черных чернил ниже, чем у синих, поэтому синие чернила лучше отличаются от печатного текста. Тем не менее, черные рукописи тоже частично распознаются.



Рисунок 17 — Пример сегментации фотографии тестового примера b) (ПРИЛОЖЕНИЯ, Рисунок 21) с рукописями, написанными синими и черными чернилами; результат сегментации выделен красным

В целом, можно заключить, что на реальных данных модели работают, но недостаточно хорошо. При этом, на отсканированных документах модели работают совсем плохо. Гипотезы, объясняющие причины низкого качества сегментации, могут быть подтверждены или опровергнуты в будущем. Подробнее про это будет сказано в заключении (Раздел 3. ЗАКЛЮЧЕНИЕ).



## **Раздел 3. ЗАКЛЮЧЕНИЕ**

### **3.1 Выводы и ключевые результаты**

Во-первых, создана система (framework) для подготовки синтетических данных в задаче сегментации рукописного текста, которая показала свою применимость (2.3.6 Результаты).

Во-вторых, было проведено исследование и получены практические результаты в виде алгоритма (модели) применимого к сегментации рукописного текста. Алгоритм показал хорошие результаты на синтезированном наборе данных (2.4.5 Результаты).

В-третьих, были рассмотрены несколько вариантов проверки качества работы алгоритма (2.5.2 Результаты и выводы) и была продемонстрирована работа алгоритма (модели) на реальных примерах.

### 3.2 Цели и задачи для дальнейшего исследования

В первую очередь в будущем стоит проверить гипотезы, почему на реальных примерах модель работает значительно хуже, чем на синтезированных данных, и добиться улучшения работы модели на реальных данных путем аугментации обучающей выборки, а именно:

- Попробовать брать фрагменты рукописного текста не только из фотографий, но и из сканов;
- Добавить к рукописям аугментации по типу grid distortion, скашивания, skew, shrink, аугментации с яркостью и контрастностью
- Попробовать брать страницы печатного текста как отсканированные, так и их фотографии.
- Провести эксперименты с методами наложения рукописей на печатный текст при генерации синтетических данных, например, попробовать альфа-смешивание (англ. alpha-blending) [Porter, Duff, 1984]

Во вторую очередь стоит качественно оценить модель на более разнообразных данных, например, взяв какой-то существующий набор данных с изображениями рукописных текстов.

По мере добавления новых аугментаций стоит заново обучать модели и проверять их на старых и новых реальных тестовых данных. Прогресс следует отслеживать и корректировать подходы.

Наконец, когда качественные результаты модели будут визуально хорошими и не будет находиться тестовых примеров, на которых модель работает плохо, стоит приступить к количественной проверке качества сегментации. Для количественной проверки нужно будет реализовать метод предложенный в (2.5.1 Методология) или альтернативный.

Также, стоит поэкспериментировать с архитектурой моделей:

1. Попробовать другие значения глубин слоев на уровнях сокращающегося и расширяющегося пути модели типа U-Net.
2. Попробовать увеличить количество уровней сокращающегося и расширяющегося пути.
3. Попробовать другие архитектуры моделей.

## СПИСОК ИСТОЧНИКОВ

1. Afzal M. Z. и др. Deepdocclassifier: Document classification with deep Convolutional Neural Network // 2015 13th International Conference on Document Analysis and Recognition (ICDAR). : IEEE, 2015.
2. Agrawal S. How to split data into three sets (train, validation, and test) And why? [Электронный ресурс]. URL: <https://towardsdatascience.com/how-to-split-data-into-three-sets-train-validation-and-test-and-why-e50d22d3e54c> (дата обращения: 13.05.2023).
3. Baidya E., Goel S. LectureKhoj: Automatic tagging and semantic segmentation of online lecture videos // 2014 Seventh International Conference on Contemporary Computing (IC3). : IEEE, 2014.
4. Belaïd A., Santosh K. C., D'Andecy V. P. Handwritten and Printed Text Separation in Real Document [Электронный ресурс]. URL: <https://arxiv.org/abs/1303.4614> (дата обращения: 14.05.2023).
5. Berman M., Triki A. R., Blaschko M. B. The Lovasz-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. : IEEE, 2018.
6. Bishop C. M. Pattern Recognition and Machine Learning. : Springer Verlag, 2006.
7. Brownlee J. A Gentle Introduction to the Rectified Linear Unit (ReLU) [Электронный ресурс]. URL: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/> (дата обращения: 17.05.2023).
8. Buslaev A. и др. Alumentations: Fast and Flexible Image Augmentations // Information. 2020. Т. 11. № 2. С. 125.

9. Contributors to Wikimedia projects. F-score [Электронный ресурс]. URL: <https://en.wikipedia.org/wiki/F-score> (дата обращения: 17.05.2023).
10. Courville A. Deep Feedforward Networks // Deep Learning. : MIT Press, 2016. С. 180–184.
11. Dansena P., Bag S., Pal R. Generation of Synthetic Data for Handwritten Word Alteration Detection // IEEE Access. 2021. Т. 9. С. 38979–38990.
12. Dice L. R. Measures of the Amount of Ecologic Association Between Species // Ecology. 1945. Т. 26. № 3. С. 297–302.
13. Franke J., Oberlander M. Writing style detection by statistical combination of classifiers in form reader applications // Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93). : IEEE Comput. Soc. Press, 1993.
14. He K. и др. Deep Residual Learning for Image Recognition // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). : IEEE, 2016.
15. Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Электронный ресурс]. URL: <https://arxiv.org/abs/1502.03167>.
16. Jo J. и др. Handwritten Text Segmentation via End-to-End Learning of Convolutional Neural Networks // Multimedia Tools and Applications. 2020. Т. 79. № 43–44. С. 32137–32150.
17. Kandan R. и др. A Robust Two Level Classification Algorithm for Text Localization in Documents // Advances in Visual Computing. Berlin, Heidelberg: Springer Berlin Heidelberg. С. 96–105.
18. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization [Электронный ресурс]. URL: <https://arxiv.org/abs/1412.6980>.

19. Koo H. I., Cho N. I. Text-Line Extraction in Handwritten Chinese Documents Based on an Energy Minimization Framework // IEEE Transactions on Image Processing. 2012. Т. 21. № 3. С. 1169–1175.
20. Li D., Wu Y., Zhou Y. Linecounter: Learning Handwritten Text Line Segmentation By Counting // 2021 IEEE International Conference on Image Processing (ICIP). : IEEE, 2021.
21. Li L. и др. Semi-Supervised Remote Sensing Image Semantic Segmentation Method Based on Deep Learning // Electronics. 2023. Т. 12. № 2. С. 348.
22. Li X.-H., Yin F., Liu C.-L. Printed/Handwritten Texts and Graphics Separation in Complex Documents Using Conditional Random Fields // 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). : IEEE, 2018.
23. Lightning team, Falcon W. Pytorch Lightning [Электронный ресурс]. URL: <https://github.com/Lightning-AI/lightning> (дата обращения: 18.05.2023).
24. Lin T.-Y. и др. Focal Loss for Dense Object Detection // 2017 IEEE International Conference on Computer Vision (ICCV). : IEEE, 2017a.
25. Lin T.-Y. и др. Focal Loss for Dense Object Detection // 2017 IEEE International Conference on Computer Vision (ICCV). : IEEE, 2017b.
26. Liu D. A Practical Guide to ReLU - Danqing Liu [Электронный ресурс]. URL: <https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7> (дата обращения: 17.05.2023).
27. Long J., Shelhamer E., Darrell T. Fully convolutional networks for semantic segmentation // 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). : IEEE, 2015.
28. Ma J. и др. Loss odyssey in medical image segmentation // ArXiv. 2020. Т. abs/2005.13449.

29. Marti U.-V., Bunke H. The IAM-database: an English sentence database for offline handwriting recognition // International Journal on Document Analysis and Recognition. 2002. T. 5. № 1. С. 39–46.

30. Paszke A., Gross S., Massa F. PyTorch: An Imperative Style, High-Performance Deep Learning Library // Advances in Neural Information Processing Systems 32 Pre-proceedings (NeurIPS 2019). : Curran Associates, Inc., 2019. С. 8024–8035.

31. Porter T., Duff T. Compositing digital images // Proceedings of the 11th annual conference on Computer graphics and interactive techniques. New York, NY, USA: ACM, 1984.

32. Radečić D. Softmax Activation Function Explained [Электронный ресурс]. URL: <https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60> (дата обращения: 17.05.2023).

33. Renton G. и др. Handwritten Text Line Segmentation Using Fully Convolutional Network // 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). : IEEE, 2017.

34. Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation // Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015. С. 234–241.

35. Ryu J., Koo H. I., Cho N. I. Language-Independent Text-Line Extraction Algorithm for Handwritten Documents // IEEE Signal Processing Letters. 2014. T. 21. № 9. С. 1115–1119.

36. Smith R. An Overview of the Tesseract OCR Engine // Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). : IEEE, 2007.

37. Souibgui M. A. и др. One-shot Compositional Data Generation for Low Resource Handwritten Text Recognition // 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). : IEEE, 2022.

38. TensorFlow Developers. TensorFlow [Электронный ресурс]. URL: <https://zenodo.org/record/4724125> (дата обращения: 19.05.2023).

39. Wikipedia contributors. Cross entropy [Электронный ресурс]. URL: [https://en.wikipedia.org/wiki/Cross\\_entropy](https://en.wikipedia.org/wiki/Cross_entropy) (дата обращения: 16.05.2023).

40. Woodie A. Deep Learning: The Confluence of Big Data, Big Models, Big Compute [Электронный ресурс]. URL: <https://www.datanami.com/2019/01/10/deep-learning-the-confluence-of-big-data-big-models-big-compute/> (дата обращения: 12.05.2023).

41. Zeiler M. D. и др. Deconvolutional networks // 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. : IEEE, 2010.

42. Zhong X., Tang J., Jimeno Yepes A. PubLayNet: Largest Dataset Ever for Document Layout Analysis // 2019 International Conference on Document Analysis and Recognition (ICDAR). : IEEE, 2019.

43. ReduceLROnPlateau [Электронный ресурс]. URL: [https://pytorch.org/docs/stable/generated/torch.optim.lr\\_scheduler.ReduceLROnPlateau.html](https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html) (дата обращения: 18.05.2023).



## ПРИЛОЖЕНИЯ

### ПРИЛОЖЕНИЕ

```
def cover_printed_with_boxes(self, printed_image, dilate=True):
    printed_grey = cv2.cvtColor(printed_image, cv2.COLOR_BGR2GRAY)
    printed_inverse = cv2.bitwise_not(printed_grey)
    printed_mask = cv2.threshold(printed_inverse,
                                255 - self.hwr_threshold,
                                255,
                                cv2.THRESH_BINARY)[1]

    handwritten_mask = np.zeros_like(printed_grey)
    overlapped_image = printed_inverse

    for i, box in enumerate(self.bboxes_iterable):
        box = box[0].numpy()
        box_grey = cv2.cvtColor(box, cv2.COLOR_BGR2GRAY)
        box_grey = self.bboxes_augmenter.transform(box_grey,
                                                    max_size=overlapped_image.shape[:2])

        box_inverse = cv2.bitwise_not(box_grey)
        box_mask = cv2.threshold(box_inverse,
                                255 - self.hwr_threshold, 255,
                                cv2.THRESH_BINARY)[1]

        if dilate:
            box_mask = cv2.dilate(box_mask,
                                  kernel=np.ones((3, 3), np.uint8),
                                  iterations=1,
                                  borderType=cv2.BORDER_ISOLATED)

        box_masked = cv2.bitwise_and(box_inverse, box_mask)

        # генерируем случайные координаты для наложения маленького
        # изображения
        h, w = overlapped_image.shape
        h_box, w_box = box_masked.shape
        x = random.randint(0, w - w_box)
        y = random.randint(0, h - h_box)

        # наложение через попиксельный максимум
        overlapped_image[y:y+h_box, x:x+w_box] = cv2.max(box_masked,
                                                            overlapped_image[y:y+h_box, x:x+w_box])
```

Программный код 4 — Код наложения рукописей на печатный текст

## ПРИЛОЖЕНИЕ

```
def create_save_boxes(self, img):  
    """  
    1. Обрезка изображения по заданной рамке (характерна для датасета)  
    2. Преобразование в градации серого  
    3. Бинаризация с подобранным вручную порогом (характерен для датасета)  
    4. Размытие изображения с применением структурирующего элемента (Erode) с характерным для  
    датасета размером ядра  
    5. Добавление белой рамки с полученному и исходному изображению  
    6. Поиск контуров для создания "рамок с текстом"  
    7. Сохранение каждого найденного фрагмента текста в специальную директорию  
    """  
  
    grey = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  
    _, thresh = cv2.threshold(grey, self.hwr_threshold, 255, cv2.THRESH_BINARY)  
    img_erode = cv2.erode(thresh, np.ones((10, 40), np.uint8), iterations=1)  
    border = 15  
    img_erode = cv2.copyMakeBorder(img_erode, border, border, border, border,  
cv2.BORDER_CONSTANT, value=(255))  
    img = cv2.copyMakeBorder(img, border, border, border, border, cv2.BORDER_REFLECT)  
    # Выделение контуров  
    contours, hierarchy = cv2.findContours(img_erode, cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)  
  
    output1 = img  
    areas = []  
    rects = []  
    for idx, contour in enumerate(contours):  
        rect = (x, y, w, h) = cv2.boundingRect(contour)  
        if hierarchy[0][idx][3] == 0:  
            rects += [rect]  
            areas += [w * h]  
  
    boxes = []  
    # Возьмём только контуры с площадью внутри квантили 20% - 90%  
    indices = np.array(areas).argsort()  
    l = len(indices)  
    indices_cropped = indices[int(0.4 * l): int(0.9 * l)]
```

Программный код 5 — Функция, выделяющая фрагменты рукописей со страницы рукописного текст и сохраняющая их в виде отдельных файлов, метод класса BoxDatasetGenerator

## ПРИЛОЖЕНИЕ

Sentence Database

A01-003

Though they may gather some Left-wing support, a large majority of Labour M Ps are likely to turn down the Foot-Griffiths resolution. Mr. Foot's line will be that as Labour M Ps opposed the Government Bill which brought life peers into existence, they should not now put forward nominees. He believes that the House of Lords should be abolished and that Labour should not take any steps which would appear to "prop up" an out-dated institution.

Though they may gather some Left-wing support, a large majority of Labour M Ps are likely to turn down the Foot-Griffiths resolution. Mr. Foot's line will be that as Labour M Ps opposed the Government Bill which brought life peers into existence, they should not now put forward nominees. He believes that the House of Lords should be abolished and that Labour should not take any steps which would appear to "prop up" an out-

Name: \_\_\_\_\_

Рисунок 18 — Пример из набора данных IAM

Research

Open Access

**A weather-driven model of malaria transmission**Moshe B Hoshen<sup>\*1</sup> and Andrew P Morse<sup>2</sup>

Address: <sup>1</sup>Virtual Population Laboratory, Department of Physics, University of Liverpool, Liverpool L69 7ZE, UK and <sup>2</sup>Department of Geography, University of Liverpool, P.O. Box 147, Liverpool, L69 3BX, UK

Email: Moshe B Hoshen<sup>\*</sup> - hoshen@liv.ac.uk; Andrew P Morse - apmorse@liv.ac.uk

<sup>\*</sup> Corresponding author

Published: 06 September 2004

Received: 04 March 2004

Malaria Journal 2004, 3:32 doi:10.1186/1475-2875-3-32

Accepted: 06 September 2004

This article is available from: <http://www.malariajournal.com/content/3/1/32>

© 2004 Hoshen and Morse; licensee BioMed Central Ltd.

This is an open-access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/2.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Abstract**

**Background:** Climate is a major driving force behind malaria transmission and climate data are often used to account for the spatial, seasonal and interannual variation in malaria transmission.

**Methods:** This paper describes a mathematical-biological model of the parasite dynamics, comprising both the weather-dependent within-vector stages and the weather-independent within-host stages.

**Results:** Numerical evaluations of the model in both time and space show that it qualitatively reconstructs the prevalence of infection.

**Conclusion:** A process-based modelling structure has been developed that may be suitable for the simulation of malaria forecasts based on seasonal weather forecasts.

**Background**

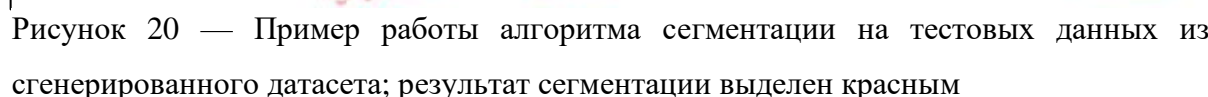
The importance of climate as a driving force of malaria transmission has been known since the earliest days of research on this devastating parasitic disease. However, it is only with the advent of effective weather forecasting techniques that this knowledge may be implemented numerically. Seasonal climate forecasting (with up to six months lead time) has developed rapidly in recent years with a number of atmospheric climate modelling groups showing evidence of skill and reliability in their systems. Because of the chaotic nature of the atmosphere, seasonal forecasts are necessarily probabilistic. These probabilistic predictions are derived from multiple integrations of deterministic climate models. These models successfully predicted the onset and demise of the 1997/1998 El Nino event and its impact on weather in Africa [1]. That event in East Africa was associated with devastating malaria epidemics [2] and, consequently, the health community has shown an increasing interest in the use of seasonal fore-

casts for predicting epidemics of climate related diseases [3].

The DEMETER project <http://www.ecmwf.int/research/demeter> was aimed to advance the concept of seasonal climate forecasts based on multi-model ensembles. The DEMETER coupled models and the DEMETER retrospective forecast (hindcast) integrations are described elsewhere [4]. The European Centre for Medium Range Weather Forecasting (ECMWF) second-generation global weather re-analysis data set ERA-40, <http://www.ecmwf.int/research/era> is being used to test the accuracy ("skill") of the hindcasts. Central to the DEMETER project is an evaluation of the potential of seasonal climate forecasts for end-user communities, such as those concerned with agricultural output and malaria epidemic control [5]. ERA is thus being used as the "gold standard" for the weather forecasts, and in the research presented

Рисунок 19 — Пример страницы из набора данных PubLayNet





## ПРИЛОЖЕНИЕ

[illegible]

Потом он прыгает, крестясь,  
в прищип, но в схватке рукопашной  
он дергает крест. Обдав себя  
в кюшете прессом втирающим

Подпись:

Подпись:

исправить!  




# ПОСЫЛКА



0155 2323

(услуга обязательной доставки посылки в выходные дни)

3333 4444

(услуга обязательного возврата посылки и возврата, руб. кол. Р)

**Заполняется отправителем**

От кого Иванов Иван Иванович

Откуда г. Дзержинский, ул. Индустриальная, пер. 6/4, кв. 111

Номер рейсового телефона +7 933 888 7166

☐ С МВЗ уполномочен

Кому Семёнову Семёну Семёновичу

Куда г. Масловская адм. р-н, г. Дзержинский, ул. Парусниковская

Номер рейсового телефона +7 933 888 7166

☐ С МВЗ уполномочен

**Заполняется оператором**



☒ Стандарт  
☒ Экспресс  
☐ Курьер EMS

☒ Нестандартная  
☐ 1 класс

Вес 11111

Отпуск К/Ш  
ОПС нести приёма

неверный 

a)

b)

*На подпись*      *05 284567 8910*

Государственное № 5  
в г.п.у.и.а. ФНС России  
от 02.10.2018 № МВБ-7-1-15580/П

**СПРАВКА О ДОХОДАХ И СУММАХ НАЛОГА ФИЗИЧЕСКОГО ЛИЦА**

за 1901 год от 01.01.1902

1. Данные о налоговом агенте

Код по ОКДТОР 1111123 Телефон 99999999 ИНН 12345678 КПП 510121

Налоговый агент ООО "Рога и копыта"

Формы организации (индивидуальный предприниматель) организация с наемными

ИНН в Российской Федерации Иванов Вид Иван Гражданство (код страны) ИВАНОВИЧ

Статус налогоплательщика \_\_\_\_\_ Дата рождения \_\_\_\_\_ Серия и номер документа \_\_\_\_\_

Код документа, удостоверяющего личность: \_\_\_\_\_

2. Данные о физическом лице - получателе дохода

3. Доходы, облагаемые по ставке \_\_\_\_\_ % исчисловый

Месяц	Код дохода	Сумма дохода	Код вычета	Сумма вычета
авг.	A21	123456		
сеп.	A65	123456		
мар.				
авр.	B1F	123456		
июл.		123456		
нояб.	B10H			
дек.	U2K6	4567		

4. Стандартные, социальные и имущественные налоговые вычеты

Код вычета	Сумма вычета	Код вычета	Сумма вычета	Код вычета	Сумма вычета	Код вычета	Сумма вычета

5. Общие суммы дохода и налога

Общая сумма дохода	Сумма налога удержанная
Налоговая база	Сумма налога уменьшенная
Сумма налога исчисленная	Сумма налога, которую удерживают налоговые агенты
Сумма фискальных зачетов (поплат)	Сумма налога, не удерживаемая налоговыми агентами

налоговый агент (в Я.О.) \_\_\_\_\_ (подпись)

\* Отчеты представляются при наличии.

Отправил в срок до 14.14.2144!

[illegible]

c)

b, 1991

d)



## ПРИЛОЖЕНИЕ

ИП Косачев Иван Иванович  
ИНН 43-04-05677

Кассовый документ  
выдаваемый на территории РФ

Серия 00

КВИТАНЦИЯ №

Квитанция № 021446  
Дата выдачи 23.03.23  
Сумма выданных  
денег (в руб.) 1111 руб.  
Получен

Заказчик

Адрес

Наименование продукции	Размер	Артикул	Номер заказа	Цвет	Фабрика/материал
кроссовки	39	1112	421	красн	
кеды					

Наименование работ и видов услуг	Единица измерения	Количество	На основании по проекту	Стоимость
кроссовки	38	1234		
кеды	40	5678		
джинсы		882		

23 446 руб.

получено при приеме товара (услуг) кроссовки  
23 446 рубли  
(получено при приеме товара (услуг) продавцом)

итого


Иванов Д.С.  
(подпись)  
Петров С.С.  
(подпись)

Заручился


(подпись)

Потом он пропал, кроссовки,  
в приходе, но в сводке руконаписной  
он чернил крест. Обозначается  
в кассовке прессом вверсионном

Подпись:



Подпись:



a)

ПОЧТА РОССИИ 

**ПОСЫЛКА**

Ф. 3-а

*исправить!*

**Заполняется отправителем**

От кого: Иванов Иван Иванович Кому: Семёнову Семёну Семёновичу

Откуда: г. Дзержинский, ул. Интернациональная, пер. 6 А, кв. 4, IIII Куда: г. Москва, ул. Масловская, д. 15, кв. 41

Номер действующего телефона: +7 999 888 1166 Адрес отправителя: г. Дзержинский, ул. Интернациональная, пер. 6 А, кв. 4, IIII

☐ С SMS уведомлением

**Заполняется оператором**



☒ Стандартный ☒ Нестандартная ☐ С описью

☐ Экспресс ☒ 1 класс

☐ Курьер EMS

Вес: 1440 г

Открыть КТД  
СГД и/или проклеить

b)

[illegible]

c)

[illegible]

d)

Рисунок 22 — Примеры сегментации фотографий заготовок для проверки (a, b, c, d) ; результат сегментации выделен красным