


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой № 52

Д.Т.Н., доцент

должность, уч. степень, звание

 01.06.17

подпись, дата

Тюрликов А.М.

инициалы, фамилия

БАКАЛАВРСКАЯ РАБОТА

на тему Программный комплекс для распознавания текстовой информации
на изображениях

выполнена Аверьяновым Сергеем Сергеевичем
фамилия, имя, отчество студента в творительном падеже

по направлению подготовки 11.03.02 Информационные
код наименование направления
технологии и системы связи

направленности 03 Программно-
код наименование направленности
защищенные инфокоммуникации
наименование направленности

Студент группы № 5321

 01.06.2017

подпись, дата

Аверьянов С.С.

инициалы, фамилия

Руководитель

доц., к.т.н.

должность, уч. степень, звание


 01.06.2017

подпись, дата

Веселов А. И.

инициалы, фамилия

В работе не содержится информация с ограниченным доступом и
отсутствуют сведения, представляющие коммерческую
ценность.

 01.06.2017 (Веселов А.И.)
Санкт-Петербург 2017

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

УТВЕРЖДАЮ

Заведующий кафедрой № 52

Д.Т.Н., доцент

должность, уч. степень, звание

 01.06.17
подпись, дата

Тюрликов А.М.

инициалы, фамилия

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ БАКАЛАВРСКОЙ РАБОТЫ

студенту группы

5321

номер

Аверьянову Сергею Сергеевичу

фамилия, имя, отчество

на тему

Программный комплекс для распознавания текстовой информации
на изображениях

утвержденную приказом ГУАП от

05.04.2017

№

07-275/17

Цель работы:

Разработка программного комплекса для распознавания текстовой информации

на изображениях

Задачи, подлежащие решению:

Исследовать и оценить возможности искусственных

нейронных сетей по распознаванию текстовой информации на изображении; разработать программный комплекс; оценить ошибку при распознавании; исследовать зависимость ошибки распознавания от параметров системы.

Содержание работы (основные разделы):

Обзор предметной области;

Система распознавания текстовой информации;


Результаты работы программы;

Срок сдачи работы « 01 » июня 2017

Руководитель

доц., к.т.н.

должность, уч. степень, звание

 06.04.2017
подпись, дата

Веселов А. И.

инициалы, фамилия

Задание принял(а) к исполнению

студент группы №

5321

 06.04.2017
подпись, дата

Аверьянов С. С.

инициалы, фамилия

Содержание

Введение	5
1 Обзор предметной области.....	6
1.1 Задача распознавания текстовой информации.....	6
1.2 Проблемы, связанные с распознаванием символов.....	7
1.3 Системы распознавания текста.....	8
1.4 Обзор методов классификации образов.....	9
1.5 Искусственные нейронные сети	10
1.5.1 Структура нейронных сетей.....	10
1.5.2 Полносвязные нейронные сети.....	12
1.5.3 Целевые критерии при обучении нейронной сети	13
1.5.4 Алгоритм обучения полносвязной нейронной сети	14
2 Система распознавания текстовой информации.....	16
2.1 Предметная область исходных данных	16
2.2 Описание модели системы распознавания текстовой информации	18
2.2.1 Предобработка изображения.....	19
2.2.2 Детектирование текста	19
2.2.3 Сегментация символов	22
2.2.4 Распознавание символов	24
2.2.5 Постобработка	25
2.3 Данные для обучения.....	26
2.3.1 Формирование выборки для детектирования текста.....	26
2.3.2 Формирование выборки для сегментации символов.....	27
2.3.3 Объединенная выборка для детектирования и сегментации.....	28
2.3.4 Формирование выборки для распознавания символов	29
2.3.5 Тестовые выборки	31
2.4 API для проектирования нейронных сетей.....	31
3 Результаты работы программы.....	36
3.1 Методика оценки качества работы разработанной системы.....	36
3.1.1 Методы оценки работы алгоритмов классификации	36

3.1.2 Критерий Accuracy	37
3.1.3 Точность и полнота	38
3.1.4 Матрица неточностей	39
3.1.5 F-score	39
3.1.6 Оценка качества распознавания текстовой информации.....	40
3.2 Оценка качества работы	41
3.2.1 Оценка качества работы классификатора, используемого на этапах детектирования и сегментации текста	41
3.2.2 Оценка качества работы детектора текста	44
3.2.3 Оценка качества работы классификатора, используемого на этапе распознавания символов	46
3.2.4 Оценка качества распознавания	50
3.3 Исследование влияния функциональных частей системы на результат	53
Заключение	57
Список литературы	58

Введение

Несмотря на то, что в настоящее время большинство документов составляется в электронном виде, задача создания полностью электронного документооборота ещё далека до полной реализации. Как правило, существующие системы охватывают деятельность отдельных организаций, а обмен данными между организациями осуществляется с помощью традиционных бумажных документов.

Задача перевода информации с бумажных на электронные носители актуальна не только в рамках потребностей, возникающих в системах документооборота. Современные информационные технологии позволяют существенно упростить доступ к информационным ресурсам, накопленным человечеством, при условии, что они будут переведены в электронный вид.

Наиболее простым и быстрым является сканирование документов с помощью сканеров. Результат работы является цифровое изображение документа – графический файл. Более предпочтительным, по сравнению с графическим, является текстовое представление информации. Этот вариант позволяет существенно сократить затраты на хранение и передачу информации, а также позволяет реализовать все возможные сценарии использования и анализа электронных документов. Поэтому наибольший интерес с практической точки зрения представляет именно перевод бумажных носителей в текстовый электронный документ.

Перевод документов, книг, рукописей в электронный вид позволяет автоматизировать различные процессы (например, бухгалтерский учет или публикацию текста на веб-странице). Такое преобразование позволяет хранить информацию в более компактной форме без потери качества, облегчает редактирование и упрощает процедуру поиска информации в тексте. Полученный текст доступен для дальнейших преобразований, например, перевод на иностранные языки или преобразование в аудио формат.

Целью данной работы является разработка программного комплекса для распознавания текстовой информации на изображениях. Для ее достижения были поставлены следующие задачи:

- 1) исследовать параметры и оценить возможности искусственных нейронных сетей по распознаванию текстовой информации на изображении;
- 2) разработать программный комплекс;
- 3) оценить ошибку при распознавании;
- 4) исследовать зависимость ошибки распознавания от параметров системы.

В первом разделе данной работы рассматривается задача распознавания текстовой информации, а также типовые трудности, возникающие при ее решении. Во втором разделе описывается предлагаемая модель системы, а также ее составные части. В третьем разделе приводится пример работы спроектированной системы, а также производится оценка качества как ее работы в целом, так и ее отдельных частей.

1 Обзор предметной области

1.1 Задача распознавания текстовой информации

Распознавание текстовой информации на изображении – это процесс преобразования графического изображения печатного или рукописного текста в текстовые данные, используемые для представления символов в цифровом виде.

Для решения задачи распознавания текстовой информации на изображении применяются методы классификации образов. Классификация решает задачу идентификации объектов, разделенных некоторым образом на классы. Каждый объект как правило описывается набором признаков, на основе которых можно отнести данный объект к одному из классов. Классифицируемые объекты принято называть образами [1]. В задаче распознавания символов в качестве классов выступают цифры и буквы алфавита, в качестве образов

изображения букв и цифр, а их признаками являются значения пикселей в этих изображениях.

В качестве исходных данных для системы распознавания выступает растровое изображение, которое содержит текстовую информацию. Желательно, чтобы изображение с текстом было как можно более высокого качества, так как зашумленность изображения и его различные искажения усложняют задачу распознавания. Для упрощения задачи распознавания может применяться предварительная обработка изображения для улучшения его качества (фильтрация шума, бинаризация и т.д.).

1.2 Проблемы, связанные с распознаванием символов

Существует набор типовых проблем, усложняющих задачу распознавания рукописного и печатного текста.

Изображение, поступающее на вход системы, может содержать символы, которые имеют различное написание. При анализе рукописного текста встречается огромное множество форм написания одних и тех же символов, в зависимости от почерка автора. Также имеется большое число различных машинных шрифтов, которые используются для печати текста, что также затрудняет их классификацию.

Растровые изображения, содержащие текст, обычно получаются в результате фотографирования или сканирования бумажного носителя. В этом случае изображения символов могут быть повернуты, зашумлены или иметь какие-либо другие искажения. Искажения могут быть вызваны некачественной печатью (пятна, непропечатывание, слипание соседних символов), эффектами освещения (блики, тени), смещением и поворотом символов в результате наклона бумажного носителя.

Помимо этого, текст, который содержится на изображении, может состоять из символов, имеющих различные размеры, что затрудняет его детектирование.

1.3 Системы распознавания текста

На данный момент существуют ряд программных комплексов, решающих задачу оптического распознавания символов. Рассмотрим некоторые из них.

CuneiForm (англ. cuneiform – клинопись), Cognitive OpenOCR — свободно распространяемая открытая система оптического распознавания текстов российской компании Cognitive Technologies.

CuneiForm позиционируется как система преобразования электронных копий бумажных документов и графических файлов в редактируемый вид с возможностью сохранения структуры и гарнитуры шрифтов оригинального документа в автоматическом или полуавтоматическом режиме. Система включает в себя две программы для одиночной и пакетной обработки электронных документов.

OCR CuneiForm была разработана компанией Cognitive Technologies как коммерческий продукт в 1993 году. Система поставлялась с наиболее популярными моделями сканеров, МФУ и ПО в России и мире: Corel Draw, Hewlet-Packard, Epson, Xerox, Samsung, Brother, Mustek, OKI, Canon, Olivetti и др. В 2008 году Cognitive Technologies открыла исходные коды OCR CuneiForm.

Tesseract (с англ. — «тессеракт») — свободная компьютерная программа для распознавания текстов, разрабатывавшаяся Hewlett-Packard с середины 1980-х по середину 1990-х, а затем 10 лет «пролежавшая на полке» [2]. В августе 2006 г. Google купил её и открыл исходные тексты под лицензией Apache 2.0 для продолжения разработки. В настоящий момент программа уже работает с UTF-8, поддержка языков осуществляется с помощью дополнительных модулей.

ABBYY FineReader — программа для оптического распознавания символов, разработанная российской компанией ABBYY.

Программа позволяет переводить изображения документов (фотографий, результатов сканирования, PDF-файлов) в электронные редактируемые форматы. В частности, в Microsoft Word, Microsoft Excel, Microsoft Powerpoint, Rich Text Format, HTML, PDF/A, searchable PDF, CSV и текстовые (plain text)

файлы. Версия 12 поддерживает распознавание текста на 190 языках и имеет встроенную проверку орфографии для 48 из них. [3]

Подобные системы, созданные для общего использования, не привязаны к одному шрифту и типу распознаваемых данных. Однако при переходе к конкретной задаче оптического распознавания текста могут появляться ошибки и точность таких систем не будет достаточной для ее решения. Если требуется обеспечить более точное распознавание определенного типа документов, то лучше использовать узконаправленную систему распознавания. В данной работе рассматривается система, предназначенная для решения задачи распознавания текста на определенном типе изображений.

1.4 Обзор методов классификации образов

Основной задачей при оптическом распознавании символов является задача классификации. При решении задачи оптического распознавания символов требуется решить несколько подзадач, связанных с классификацией объектов, таких как обнаружение областей с текстом (бинарная оценка наличия текста в определенной области изображения), разделение текста на отдельные символы, а также распознавание найденных символов на изображении, что связано с принятием решения о принадлежности символа к соответствующему классу.

Существует множество методов классификации образов [1], таких как:

- сравнение с эталоном;
- линейный классификатор;
- искусственные нейронные сети.

Искусственные нейронные сети являются передовой технологией в области классификации изображений. В 2012 году состоялся конкурс ImageNet, посвященный классификации объектов на изображениях, содержащих образы, принадлежащие 1000 различных классов. Команда Джеффри Хинтона разработала алгоритм, основанный на искусственных нейронных сетях, который

смог победить в данном конкурсе с отрывом в два раза от ближайшего результата [4].

В данной работе в качестве метода классификации используются искусственные нейронные сети, так как они позволяют достигать высокого качества принятия решения о принадлежности объекта к одному из распознаваемых классов.

1.5 Искусственные нейронные сети

1.5.1 Структура нейронных сетей

Искусственные нейронные сети (ИНС) представляют собой систему соединённых и взаимодействующих между собой искусственных нейронов. Рассмотрим модель простейшего искусственного нейрона.

Нейрон состоит из входов, ядра нейрона и выхода. Каждый вход имеет вещественный вес, который определяет, насколько соответствующий вход нейрона влияет на его состояние. Ядро нейрона осуществляет обработку входных сигналов и формирует выходной сигнал. Выход связывает нейрон с нейронами следующего слоя. На рисунке 1 представлена схема искусственного нейрона, на которой x_i – значение i -го входа нейрона, w_i – вес i -го входа нейрона, σ – ядро нейрона, вычисляющее значение активационной функции $\sigma(z)$, a – состояние нейрона, то есть его выходной сигнал, который вычисляется по формуле 1, z – взвешенная сумма входов нейрона, которая вычисляется по формуле 2.

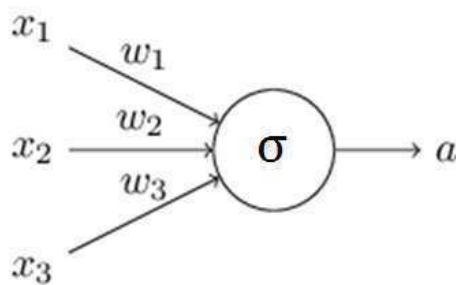


Рисунок 1 – Схема искусственного нейрона

$$a = \sigma(z), \quad (1)$$

$$z = \sum_{j=1}^n w_j x_j + b, \quad (2)$$

где n – количество входов, w_i – вес входа (связи), x_i – значение i -го входа нейрона, b – смещение нейрона.

В качестве активационной функции может использоваться сигмоид:

$$\sigma(x) = \frac{1}{1 + e^{-ax}}. \quad (3)$$

Основное достоинство этой функции в том, что она дифференцируема на всей области определения и имеет простую производную:

$$\sigma'(x) = a\sigma(x)(1 - \sigma(x)). \quad (4)$$

Данная функция принимает значения от 0 до 1 и имеет вид, представленный на рисунке 2.

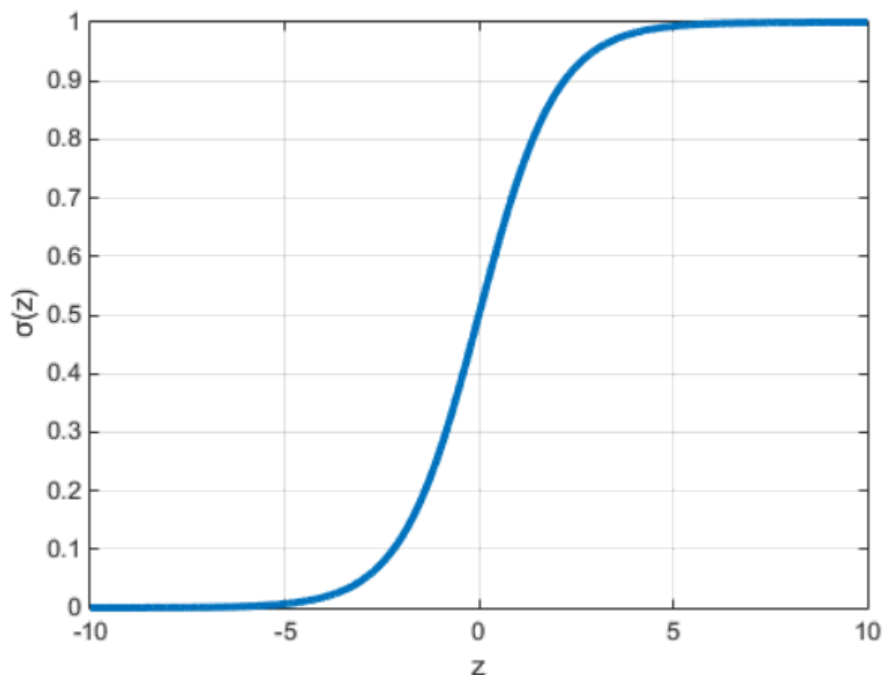


Рисунок 2 – График сигмоидальной активационной функции

Одно из главных преимуществ алгоритмов машинного обучения, в том числе и нейронных сетей, перед традиционными алгоритмами – это возможность обучения. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна

выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных или частично искаженных данных.

1.5.2 Полносвязные нейронные сети

В данной работе используется полносвязная нейронная сеть прямого распространения с обратным распространением ошибки. В полносвязной сети каждый нейрон в каждом слое связан с каждым нейроном следующего слоя. Сигнал распространяется в направлении от входного слоя к выходному. При обучении ошибка распространяется от выходного слоя к входному, то есть в направлении, противоположном направлению распространения сигнала.

Полносвязная сеть состоит из нескольких слоев: входной слой, один или несколько скрытых слоев и выходной слой [5]. Количество входных нейронов соответствует количеству пикселей в изображениях, которые поступают на вход [6]. Количество нейронов в выходном слое соответствует количеству классов, которые могут быть распознаны нейронной сетью. Количество скрытых слоев и нейронов в них выбирается экспериментально. Пример схемы 4-х слойной полносвязной нейронной сети с одним выходом представлен на рисунке 3.

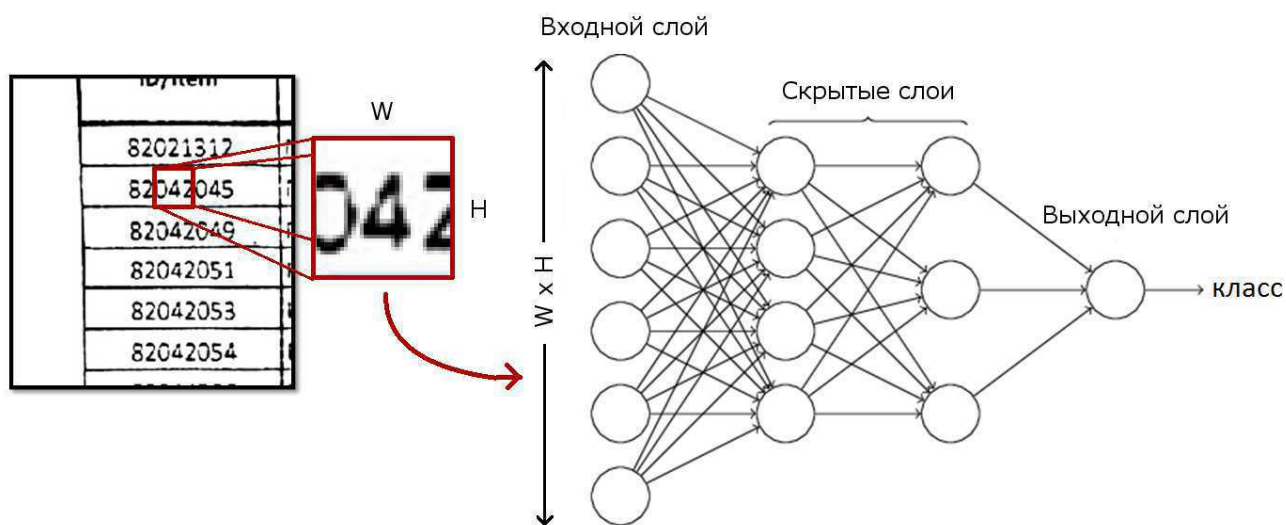


Рисунок 3 – Схема полносвязной нейронной сети, состоящей из 4 слоев

Значения выходов нейронов каждого слоя могут быть получены по формуле:

$$a_j^{(l)} = \sigma \left(\sum_{k=1}^n w_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)} \right), \quad (5)$$

где l – номер слоя, j – порядковый номер нейрона в слое l , k – порядковый номер нейрона в слое $l - 1$, n – количество нейронов в предыдущем слое, то есть в слое $l - 1$.

1.5.3 Целевые критерии при обучении нейронной сети

Задача обучения нейронной сети сводится к нахождению функциональной зависимости $Y=f(X)$, где X – входной, а Y – выходной вектор. При обучении ставится задача минимизации целевой функции ошибки нейронной сети [5].

В качестве функции ошибки в задаче классификации как правило используют функцию кросс-энтропии, которая имеет следующий вид:

$$C = -\frac{1}{n} \sum_{i=1}^N \sum_{j=1}^{J^L} \left[y_{ij} \ln a_j^{(L)} + (1 - y_{ij}) \ln (1 - a_j^{(L)}) \right], \quad (6)$$

где N – размер тренировочной выборки X , a – выход нейронной сети при входных данных x из тренировочной выборки X , y_{ij} – желаемый результат, соответствующий входным данным x_{ij} , $J^{(l)}$ – количество нейронов в слое l .

Обучение нейронной сети производится методом стохастического градиентного спуска [7]. Изменение весов связей производится по формуле:

$$\Delta w_{jk}^{(l)} = -\eta \cdot \frac{\partial C}{\partial w_{jk}^l}, \quad (7)$$

где η – параметр, определяющий скорость обучения. Параметр η выбирается методом подбора. При больших его значениях нейронная сеть обучается быстрее, однако при низких значениях предельная точность вычисления может оказаться выше, но потребуются больше эпох для обучения.

Также на каждой итерации вычисляются величины, на которые будут изменяться смещения нейронов:

$$\Delta b_j^{(l)} = -\eta \cdot \frac{\partial C}{\partial b_j^{(l)}}. \quad (8)$$

Введем переменную $\delta_j^{(l)}$, обозначив ею величину ошибки. Для k -го нейрона слоя l данная величина будет равна:

$$\delta_k^{(l)} = \sum_{j=1}^{J^{(l+1)}} \left(w_{jk}^{(l+1)} \delta_j^{(l+1)} \right) \cdot \sigma' \left(z_k^{(l)} \right). \quad (9)$$

Так как при обучении известен целевой выходной вектор, соответствующий входным данным, то для j -го нейрона выходного слоя величина ошибки будет равна:

$$\delta_j^{(L)} = \left(a_j^{(L)} - y_{ij} \right), \quad (10)$$

где L – номер выходного слоя нейронной сети.

Значение ошибки весов связей и смещений нейронов вычисляются по следующим формулам:

$$\frac{\partial C}{\partial w_{jk}^{(l)}} = a_k^{(l-1)} \delta_j^{(l)} \quad (11)$$

$$\frac{\partial C}{\partial b_j^{(l)}} = \delta_j^{(l)} \quad (12)$$

1.5.4 Алгоритм обучения полносвязной нейронной сети

Для обучения нейронной сети требуется для всей тренировочной выборки выполнить следующую последовательность шагов.

- 1) Подать на вход нейронной сети пример x_i из обучающей выборки X .
- 2) Для каждого слоя $l = 2, 3, \dots, L$ вычислить: $a_j^{(l)} = \sigma \left(\sum_{k=1}^n w_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)} \right)$.
- 3) Для каждого нейрона $j = 1, 2, \dots, J^{(L)}$ вычислить вектор ошибки:

$$\delta_j^{(L)} = \left(a_j^{(L)} - y_{ij} \right).$$

- 4) Для каждого k -го нейрона слоя $l = L-1, L-2, \dots, 2$ вычислить:

$$\delta_k^{(l)} = \sum_{j=1}^{J^{(l+1)}} \left(\left(w_{jk}^{(l+1)} \right) \delta_j^{(l+1)} \right) \cdot \sigma' \left(z_k^{(l)} \right).$$

5) Для каждого слоя $l = L, L-1, \dots, 2$ изменить значения весов связей и смещений по правилу:

$$w_{jk}^{(l)} \rightarrow w_{jk}^{(l)} - \frac{\eta_q}{m} \delta_j^{(l)} a_k^{(l-1)},$$

$$b_j^{(l)} \rightarrow b_j^{(l)} - \frac{\eta_q}{m} \delta_j^{(l)},$$

где m – размер тренировочной выборки, η_q – скорость обучения, q – индекс эпохи обучения.

2 Система распознавания текстовой информации

2.1 Предметная область исходных данных

В качестве набора исходных изображений с текстом были выбраны отсканированные сметы поставок деталей вагонов метро. Данные на изображении представляют собой таблицу, содержащую цифры, а также буквы кириллицы и латиницы. На рисунке 4 представлен пример изображения выбранной предметной области.

ID/Item	Description	Description CZ	MJ	Qty	Invoice	TOTAL netto kg	TOTAL price CZK
8201312	МИКРОФОН	MIKROFON VLAKOVÉHO HOZHLASU	ks	1	P033	0,20	2 376,30
82041045	ПАНЕЛЬ	PANEL INFORMAČNÍ-ČLOVĚ STANIC	ks	1	P033	27,50	71 934,50
82042049	ПАНЕЛЬ	PANEL INFORMAČNÍ VNITŘNÍ	ks	4	P033	15,68	81 576,20
82042051	КОММУТАТОР	PŘEVODNÍK KOMUNIKACE	ks	1	P033	4,00	3 195,74
82042053	ИНТЕРКОМ	INTERKOM	ks	2	P033	2,64	21 973,74
82042054	БЛОК	BLOK SPOJENÍ SOUPRAV	ks	1	P033	3,60	25 023,60
82045056	ПАНЕЛЬ	PANEL INFORMAČNÍ	ks	8	P033	90,40	189 945,76
82045545	УПРАВЛЯЮЩИЙ МОДУЛЬ САРМАТ	JEDNOTKA ŘIDIČI SARMAT	ks	1	P033	4,73	116 285,10
82045646	ТАБЛО	ZOBRAZOVACÍ OKNA OBĚH-U	ks	1	P038	5,10	34 602,46
82049849	ДАТЧИК УПО	SNÍMAČ UPO	ks	1	P033	0,20	11 555,74
82050032	ИНТЕРКОМ ЛЕВЫЙ	INTERKOM PRO INVALIDY LEVÝ	ks	1	P033	1,91	10 936,87
82050045	ИНТЕРКОМ ПРАВЫЙ	INTERKOM PRO INVALIDY PRAVÝ	ks	1	P033	1,30	10 936,87
82021320	АНТЕННА РАДИОСТАНЦИИ	ANTÉNA RADIOSTANICE	ks	1	P034	1,20	1 258,55
D0585754	ЯДРИК КЛЕММНИКА X38-8 В СБОРЕ	KLADICE X38-8	ks	1	P035	21,16	6 192,51
D0578529	ЛИСТ ОБЩИВЫ	DESKA OBLOŽENÍ	ks	2	P036	0,06	5 047,62
D0581173	ЛИСТ ОБЩИВЫ	DESKA OBLOŽENÍ	ks	1	P036	5,61	4 184,78
D0581174	ЛИСТ ОБЩИВЫ	DESKA OBLOŽENÍ	ks	1	P036	5,61	3 135,74
D0578579	ЛИСТ ОБЩИВЫ	DESKA OBLOŽENÍ	ks	6	P036	38,37	15 142,86
82021322	НАУШНИК	SLUCHÁTKO RADIOSTANICE	ks	1	P038	0,39	4 690,23
82021323	ПУЛЬТ	PULT ŘÍZENÍ RADIOSTANICE	ks	1	P038	1,60	58 627,87
82023022	АНТЕННА РАДИОСТАНЦИИ	ANTÉNA RADIOSTANICE	ks	1	P038	1,10	3 808,32
82033042	ВЫКЛЮЧАТЕЛЬ	SPÍNAČ TLAKOVÝ	ks	1	P038	1,50	1 900,37
82038321	РЕГУЛЯТОР	UPRAVOVAČ TLAKU	ks	1	P038	0,26	3 476,24
82038333	РЕГУЛЯТОР	REGULÁTOR PRŮTOKU JEDNOSMĚRNÝ	ks	1	P038	0,22	276,72
82041437	АВТОСТОП	AUTOSTOP	ks	1	P038	9,80	9 575,85
82041746	РЕПРОДУКТОР	REPRODUKTOR	ks	2	P038	0,64	484,66
82047104	ШЛАНГ	HADICE AUTOSTOPU	ks	1	P038	1,93	814,54
82042881	РЕПРОДУКТОР	REPRODUKTOR	ks	1	P038	0,31	38,63
82043528	БЛОК АНТЕННО-СОГЛАСУЮЩЕГО УСТРОЙ	BLOK ANTÉNNÍHO PRÍPOSOBOVACÉ	ks	1	P038	4,00	42 993,77
82043529	БЛОК	BLOK RADIOSTANICE	ks	1	P038	7,70	53 026,96
82050146	ПРОВОД	VOČÍK	m	125	P038	7,88	2 572,30
82042159	ЛИСТ ФАНЕРНЫЙ	DESKA PŘEKLIŽOVANÁ	ks	17	P038	918,00	66 297,96
D0581174	ЛИСТ ОБЩИВЫ	DESKA OBLOŽENÍ	ks	2	P039	31,23	6 586,32
82049666	СТЕКЛО КАБИНЫ БОКОВОЕ	SKLO KABINY BOČNÍ	ks	1	P039	5,00	13 758,01
82049954	РЕЗЕРВУАР P10-100	VZDUCHOJEM P10-100	ks	3	P039	120,00	14 399,01
82049995	РЕЗЕРВУАР P10-20	VZDUCHOJEM P10-20	ks	1	P039	10,00	2 669,52
82049996	РЕЗЕРВУАР P10-9	VZDUCHOJEM P10-9	ks	1	P039	6,00	2 669,52
D0578418	ЛИСТ ОБЩИВЫ	DESKA OBLOŽENÍ	ks	1	P039	4,22	4 146,94
D0578419	ЛИСТ ОБЩИВЫ	DESKA OBLOŽENÍ	ks	1	P039	7,50	6 883,08
D0578430	ЛИСТ ОБЩИВЫ	DESKA OBLOŽENÍ	ks	1	P039	3,48	53 698,44
D0581173	ЛИСТ ОБЩИВЫ	DESKA OBLOŽENÍ	ks	2	P039	31,23	8 262,62
82042693	ПАТРОН, КОЛБА ОСЛ-1	PATRONA HASIČÍ	ks	4	P040	4,80	1 719,76
82020647	СЧЕТЧИК УЧЕТА ЭЛЕКТРОЭНЕРГИИ	ELEKTROMĚR	ks	1	P040	2,06	13 249,24
82050548	БАЛЛАСТНОЕ УСТРОЙСТВО ПИТАНИЯ	PŘÍDRÁŽNÍK	ks	1	P040	3,00	496,85
82050549	ПРЕОБРАЗОВАТЕЛЬ ИНТЕРФЕЙСА EIA-485	PŘEVODNÍK EIA-485/CAN	ks	1	P040	3,00	2 981,08
82049637	ОКНО	OKNO MALÉ LEVÉ	ks	1	P041	18,00	7 575,30
82049638	ОКНО	OKNO MALÉ PRAVÉ	ks	1	P041	18,00	7 575,30
82049645	ОКНО	OKNO	ks	2	P041	70,60	19 713,04
82042159	ЛИСТ ФАНЕРНЫЙ	DESKA PŘÍ KLÍŽOVANÁ	ks	6	P041	432,00	81 199,04
82049711	ПАНЕЛЬ ПОТОЛОЧНАЯ	PANEL STROPNÍ	ks	1	P042	17,52	2 042,99
82049712	ПАНЕЛЬ ПОТОЛОЧНАЯ	PANEL STROPNÍ	ks	3	P042	66,12	27 362,79
82049713	ПАНЕЛЬ ПОТОЛОЧНАЯ	PANEL STROPNÍ	ks	2	P042	44,08	3 951,76
82049714	ПАНЕЛЬ ПОТОЛОЧНАЯ	PANEL STROPNÍ	ks	1	P042	8,20	2 491,59
82049715	ПАНЕЛЬ ПОТОЛОЧНАЯ	PANEL STROPNÍ	ks	1	P042	22,20	3 292,93
82021315	ДАТЧИК ПОЖАРНОЙ СИГНАЛИЗАЦИИ	ČIDLO POŽÁRNÍ SIGNALIZACE	ks	11	P043	0,88	9 501,58
82021325	ПУЛЬТ	PULT OVLÁDACÍ	ks	1	P043	1,60	30 882,04
82021326	УПРАВЛЯЮЩИЙ МОДУЛЬ	JEDNOTKA ŘIDIČI	ks	1	P043	1,60	30 196,48
82042694	ИНДИКАТОР	INDIKÁTOR HASIČÍ PATRONY	ks	4	P043	0,80	1 563,40
82050528	РАСПРЕДЕЛИТЕЛЬНАЯ КОРОБКА (ПЛАТА)	SVORKOVNICE ROZBŮSOVACÍ	ks	11	P043	4,40	38 576,12
82050546	БЛОК	BLOK PRO PŘENOS INFORMACÍ	ks	1	P043	1,20	122 075,41
82050047	РЕГИСТРАТОР	REGISTRÁTOR	ks	1	P043	5,50	208 627,33
82050656	КОМПЛЕКТ МОНТАЖНЫЙ №10	PŘÍSLUŠENSTVÍ Č.10	ks	1	P043	0,01	445,76
82050657	КОМПЛЕКТ МОНТАЖНЫЙ №9	PŘÍSLUŠENSTVÍ Č.9	ks	1	P043	0,01	684,12
82050659	КОМПЛЕКТ МОНТАЖНЫЙ №8	PŘÍSLUŠENSTVÍ Č.8	ks	3	P043	0,03	1 243,56
82050660	КОМПЛЕКТ МОНТАЖНЫЙ №7	PŘÍSLUŠENSTVÍ Č.7	ks	1	P043	0,01	310,96

Рисунок 4 – Пример изображения, содержащего текст

Опишем основные особенности, присущие данному набору изображений. На таких изображениях текст преимущественно одного размера. Символы всегда

черного цвета на белом фоне, что обеспечивает высокую контрастность изображения. Однако задачу распознавания затрудняет наличие небольших пространственных искажений и наличие шума квантования спектральных коэффициентов, связанных с тем, что изображения хранятся в формате JPEG. На рисунках 5, 6 и 7 представлены различные искажения, встречающиеся на изображениях выбранной предметной области.

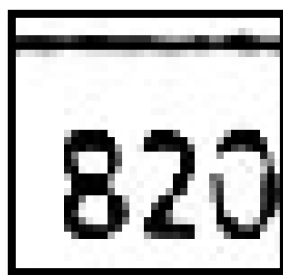


Рисунок 5 – Шум квантования спектральных коэффициентов на изображении

Также на изображениях присутствуют дефекты печати в виде плохо пропечатанных областей и пятен.



Рисунок 6 – Точечные пятна на изображении

40	4,80	1 7
40	2,00	13 24
40	3,00	49
40	3,00	2 98
41	18,00	7 57
41	18,00	7 57
41	70,00	19 71
41	432,00	31 19

Рисунок 7 – Оттиск печати на фоне текста

Для улучшения точности распознавания требуется учитывать возможные искажения на исходных изображениях и по возможности компенсировать их предварительной обработкой изображения перед распознаванием.

2.2 Описание модели системы распознавания текстовой информации

В данной работе предлагается использовать алгоритм распознавания текстовой информации, схема которого представлена на рисунке 8.

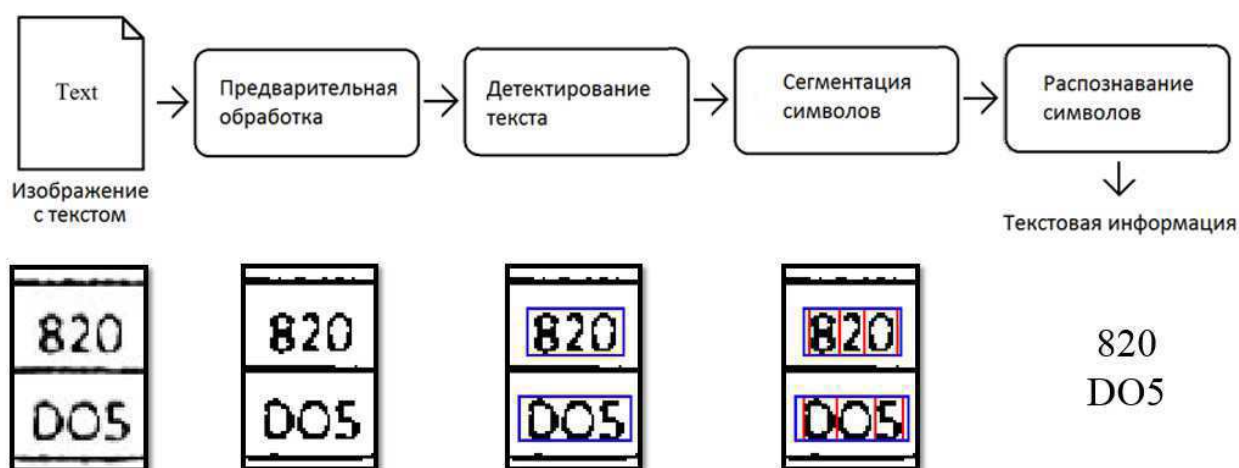


Рисунок 8 – Алгоритм работы системы распознавания текстовой информации

Алгоритм состоит из следующих этапов, выполняющихся последовательно:

- 1) предварительная обработка изображения, улучшающая качество исходного изображения;
- 2) детектирование текста, то есть обнаружение областей на изображении, содержащих текст;
- 3) сегментация областей с текстом на отдельные символы;
- 4) распознавание отдельных символов и формирование текстовой последовательности.

На вход системы подается изображение, содержащее текст. Предполагается, что текст на изображении расположен блоками, так что можно выделить области, где размер символов и расстояние между ними изменяется не

сильно. Символы в этих областях располагаются последовательно и формируют строки.

На выходе системы формируется текстовая последовательность, являющаяся результатом распознавания.

Рассмотрим назначение и работу основных блоков разработанного алгоритма подробнее.

2.2.1 Предобработка изображения

К исходному изображению, поступившему на вход системы, могут применяться предварительные преобразования, позволяющие улучшить качество этого изображения и облегчающие задачи последующих этапов.

На данном этапе могут применяться различные преобразования, такие как:

- 1) фильтрация изображения для минимизации шума (медианный фильтр, фильтр Гаусса, билатеральный фильтр);
- 2) бинаризация изображения для уменьшения количества различных значений светимости пикселей;
- 3) повышения резкости и контрастности исходного изображения;
- 4) выравнивание изображения для уменьшения искажений, связанных с поворотами и перспективой.

Из-за высокой контрастности исходных изображений, слабых пространственных искажений, а также слабого влияния шума этапом предобработки в данной задаче можно пренебречь.

2.2.2 Детектирование текста

Задачей данного этапа алгоритма является выделение областей, содержащих текстовую информацию. Для детектирования текста используется полносвязная искусственная нейронная сеть, состоящая из трех слоев, обученная на выборке, содержащей изображения букв и цифр, а также набор изображений, на которых букв и цифр нет. Количество нейронов в входном слое соответствует

размеру изображения (17 x 17). Количество нейронов в скрытом слое выбиралось экспериментально и равно 250. Выход нейронной сети показывает вероятность наличия символа в окне.

Выделение областей с текстом производится в несколько шагов.

- 1) Входное изображение, пример которого представлен на рисунке 9, обрабатывается методом скользящего окна с использованием спроектированной нейронной сети.

Если выход нейронной сети больше определенного порога, то вся область в текущем окне помечается как область, которая может содержать текст. Порог бинаризации вычисляется методом Оцу [8]. Суть метода заключается в разделении пикселей на изображении на классы полезных и фоновых пикселей, после чего порог бинаризации выбирается таким образом, чтобы внутриклассовая дисперсия была минимальной. В результате получается карта вероятностей наличия текста на изображении, пример которой представлен на рисунке 10.

82050661	1	PO43	0,01	358,80
82050662	1	PO43	0,01	478,40
82050663	1	PO43	0,01	496,13
82050664	1	PO43	0,01	520,05
82050665	8	PO43	0,08	2321,12
82050666	1	PO43	0,01	713,95
82050667	1	PO43	0,01	477,00
82050668	1	PO43	0,01	281,70

Рисунок 9 – Пример исходного изображения, содержащего как четкий текст без пространственных искажений и с высоким качеством печати (столбец слева), так и искаженный текст (4 правых столбца), содержащий плохо пропечатанные символы

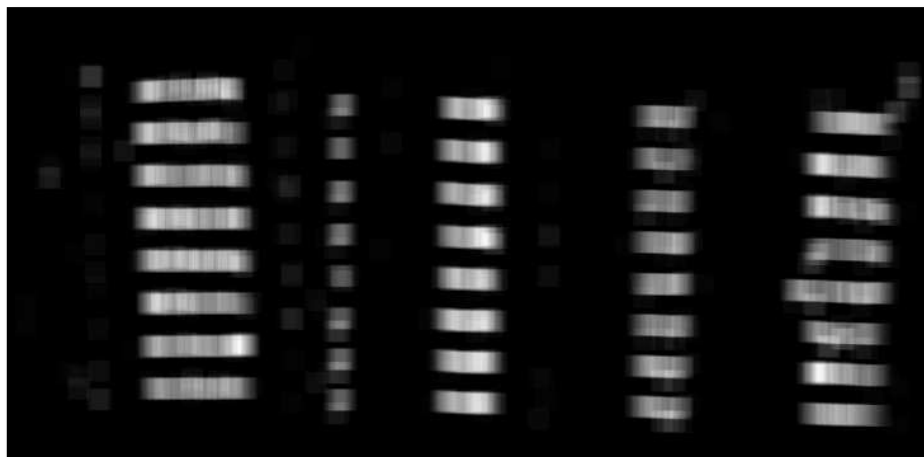


Рисунок 10 – Карта вероятностей наличия текста на изображении

- 2) Полученная карта вероятностей подвергается бинаризации по порогу для фильтрации областей с низкой вероятностью наличия текста, и формируется бинарная маска текста, пример которой представлен на рисунке 11.



Рисунок 11 – Маска текста на изображении

- 3) Далее для удобства обработки результатов детектирования производится преобразование полученных областей с текстом в области прямоугольной формы. Однако оригинальная маска текста не удаляется, так как она потребуется в дальнейшем. Пример результата данного преобразования представлен на рисунке 12.

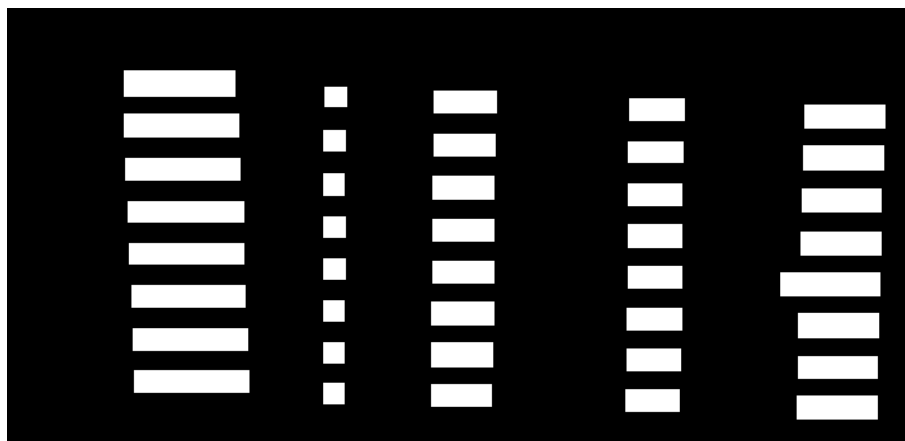


Рисунок 12 – Маска текста на изображении (прямоугольные области)

На выходе детектора получается бинарная маска, размерами совпадающая с исходным изображением, на которой выделены области, содержащие текст.

2.2.3 Сегментация символов

На этапе сегментации символов производится поиск границ символов внутри областей с текстом, что позволяет выделить области, содержащие отдельные буквы и цифры. Для решения этой задачи используется полносвязная искусственная нейронная сеть, состоящая из трех слоев, выход которой показывает вероятность наличия межсимвольного интервала. Количество нейронов в входном слое соответствует размеру изображения (17 x 17). Количество нейронов в скрытом слое выбиралось экспериментально и равно 250. Обучение нейронной сети производилось на выборке, включающей в себя изображения межсимвольных интервалов и изображений символов.

Сегментация текста производится в несколько шагов.

- 1) Входное изображение, обрабатывается методом скользящего окна с использованием данной нейронной сети аналогичным предыдущему этапу образом. В результате получается карта вероятностей наличия межсимвольных интервалов на изображении, пример которой представлен на рисунке 13.

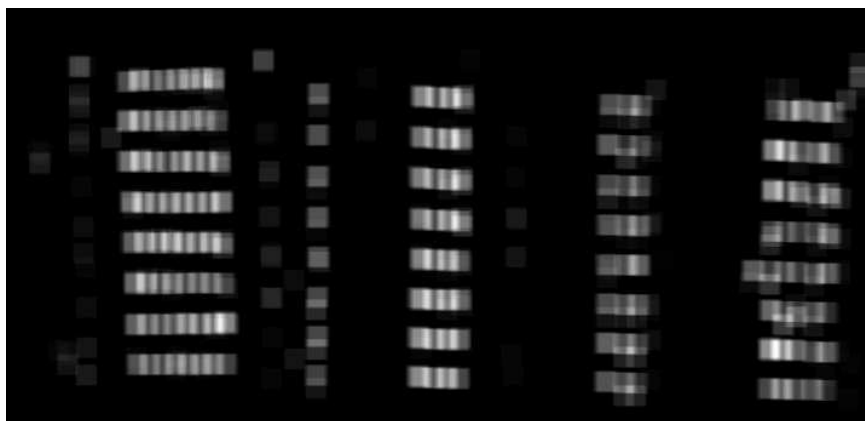


Рисунок 13 – Карта вероятностей наличия межсимвольных интервалов на изображении

Так как данный этап функционально похож на первый этап обработки изображения детектором текста, их можно объединить, вычисляя карту вероятностей наличия текста и карту вероятностей наличия межсимвольных интервалов за один цикл обработки изображения.

- 2) Затем на карте вероятностей наличия межсимвольных интервалов производится поиск локальных максимумов. Данная процедура производится методом скользящего окна. В окне производится поиск максимальных значений, которые записываются как результат. Значения остальных пикселей заменяются нулями. Пример результата такого преобразования представлен на рисунке 14. В данной работе применяется окно размером 7 пикселей в высоту и ширину. Такие размеры обоснованы размером символов на изображении (предполагается, что символ на изображении не может быть меньше чем 7 пикселей в ширину).

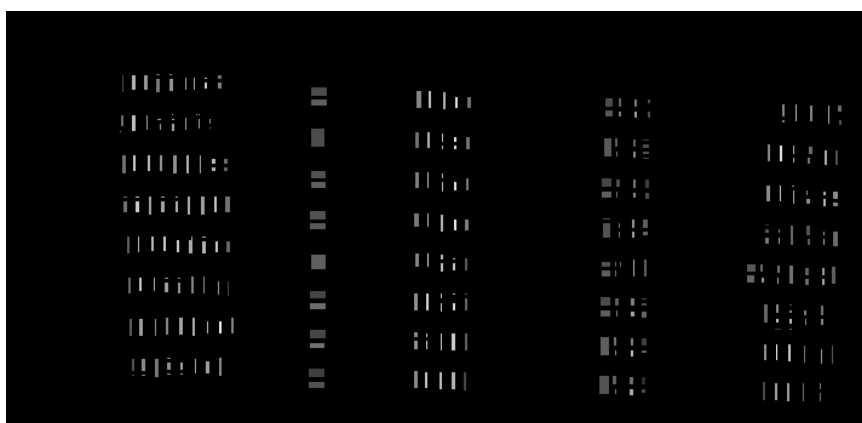


Рисунок 14 – Результат работы алгоритма поиска локальных максимумов

Для удобства дальнейшей обработки данный результат подвергается бинаризации.

- 3) Полученный результат поступает на этап фильтрации ошибочных откликов нейросети, которые оказываются за пределами полученной ранее маски текста. Также на изображении производится переформирование межсимвольных интервалов в прямые линии, высота которых определяется высотой области, содержащей текст, в которой данный межсимвольный интервал был найден. Пример работы данного этапа представлен на рисунке 15.

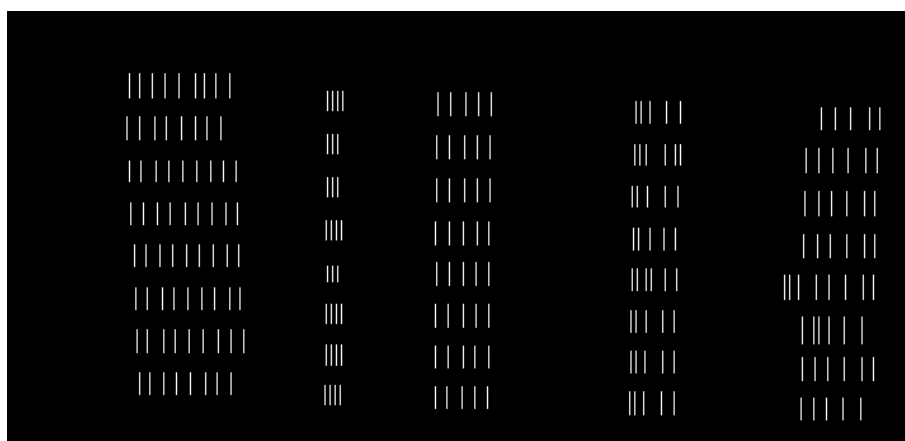


Рисунок 15 – Результат работы алгоритма поиска локальных максимумов. Результатом является маска межсимвольных интервалов, размерами совпадающая с исходным изображением.

2.2.4 Распознавание символов

На вход этапа распознавания символов поступают результаты, полученные на предыдущих этапах. Данный модуль разделяет текст на слова, распознает отдельные буквы и формирует текст.

На данном этапе также используется полносвязная нейронная сеть, состоящая из трех слоев. Размер входного слоя определяется аналогичным образом, то есть количество нейронов в нем равно количеству пикселей в окне (17 x 17). Количество выходов соответствует количеству различных букв и цифр, которые она была обучена классифицировать. Скрытый слой содержит 500

нейронов. Это обусловлено большим количеством классов чем на предыдущих этапах.

Используя маску текста и маску межсимвольных интервалов формируется набор изображений для каждой обнаруженной области с текстом. На каждом изображении содержится только один символ. Межсимвольные интервалы принимаются за границы символов. Такие изображения формируются путем расположения в окне двух соседних границ (межсимвольных интервалов) таким образом, чтобы они были равноудалены от центра. Такой способ установки окна позволяет отцентрировать найденный символ посередине окна по горизонтали. Также используется оригинальная маска областей с текстом, верхняя и нижняя граница в окне которой принимается за верхнюю и нижнюю границу символа, что позволяет отцентрировать символ в окне по вертикали.

Полученные изображения символов подаются на вход нейронной сети для определения класса, к которому они принадлежат, то есть распознается символ, изображенный посередине окна.

Выходом сети является номер класса, соответствующий определенному символу (букве или цифре). Символы записываются последовательно в строку и разбиваются пробелами на слова на основе маски областей с текстом.

2.2.5 Постобработка

Далее может применяться этап постобработки распознанного текста, который служит для исправления возможных ошибок распознавания, однако в рамках данной работы этот этап не используется.

Например, если распознанная последовательность символов, разделенная пробелами, преимущественно состоит из букв, то вероятно наличие в данном слове цифр свидетельствует об ошибке, так как некоторые цифры имеют похожее написание (например, буква О и цифра 0). Также возможны ошибки, связанные с похожим написанием букв кириллицы и латиницы.

Для улучшения качества распознанного текста по каждому полученному слову может производиться поиск в заранее подготовленном словаре. Если слово

не было найдено в словаре, то производится поиск наиболее похожего слова и предполагаемая ошибка исправляется.

2.3 Данные для обучения

Формирование тренировочной выборки имеет принципиально важное значение для успешного решения задачи машинного обучения [9]. На качество выборки влияет несколько факторов.

Тренировочная выборка должна быть репрезентативной, то есть должна достоверно и полно отражать признаки, характерные общей совокупности распознаваемых образов.

Тренировочная выборка не должна содержать противоречий, так как нейронная сеть однозначно сопоставляет выходные значения входным [10].

В данной работе используется несколько нейронных сетей и для их обучения требуется несколько отдельно подготовленных тренировочных выборок, соответствующих задачам, которые данные нейронные сети предназначены решать. Формирование данных для тренировочной выборки производилось вручную с использованием части изображений, входящих в состав выбранной предметной области. Данный процесс является трудоемким и занимает большое количество времени, так как требует участия человека.

2.3.1 Формирование выборки для детектирования текста

Данная выборка состоит из двух частей: положительной и отрицательной. Отрицательной частью выборки называется множество образов, входящих в состав общей выборки, принадлежащих к нулевому классу, а положительной частью, соответственно, множество объектов, принадлежащих к первому классу.

В положительной части выборки содержатся изображения букв и цифр, расположенных посередине изображения. Каждое изображение содержит один символ, однако по краям могут встречаться фрагменты соседних символов.

Представленные на этих изображениях символы имеют одинаковые размеры и расположены всегда на одинаковом расстоянии от краев изображения.

В отрицательной части выборки содержатся изображения аналогичного размера, не содержащие символа целиком. На таких изображениях может встречаться шум, фрагменты границ таблицы, фрагменты символов, пустоты.



Рисунок 16 – Пример изображений из положительной выборки (цифры)



Рисунок 17 – Пример изображений из отрицательной выборки

Каждое значение пикселя лежит в диапазоне от 0 (представляет белый цвет) до 255 (представляет черный цвет). Промежуточные значения отражают оттенки серого.

Для увеличения размера тренировочной выборки к оригинальным изображениям применяются аффинные преобразования и полученные новые изображения также включаются в выборку.

2.3.2 Формирование выборки для сегментации символов

Выборка для сегментации также состоит из двух частей:

- 1) положительная выборка, содержащая изображения, на которых присутствует межсимвольный интервал;
- 2) отрицательная выборка, которая содержит пустые изображения, соответствующие пробелам, а также изображения символов.

С помощью данной выборки нейронная сеть может быть обучена детектировать границы между символами.

2.3.3 Объединенная выборка для детектирования и сегментации

Так как этапы вычисления карты вероятностей наличия текста и карты вероятностей наличия межсимвольных интервалов функционально похожи и выполняются за один цикл обработки изображения, то возможно объединить две нейронные сети, использующиеся на данных этапах, в одну. Для обучения такой нейронной сети используется объединенная выборка детектирования и сегментации.

Такая выборка разделяется на три части (3 класса соответственно):

- 1) изображения, которые не содержат отцентрированных символов и межсимвольных интервалов;
- 2) изображения, которые содержат символы, расположенные посередине;
- 3) изображения, которые содержат межсимвольные интервалы, расположенные посередине.

Таким образом можно обучить нейронную сеть одновременно для выполнения двух задач: детектирования текста и его сегментации.

После формирования выборки необходимо оценить ее качество. Для этого используется алгоритм t-SNE, который позволяет визуализировать данные, переводя их из пространства высокой мерности (в данном случае из пространства размерностью 17 x 17) в пространство меньшей мерности (в 3D или 2D). Точки обозначают объекты выборки, а их цвет показывает принадлежность этого объекта к определенному классу. Если точки одного цвета группируются в пространстве, то существует возможность их разделить.

На рисунке 18 представлена визуализированная выборка для детектирования и сегментации.

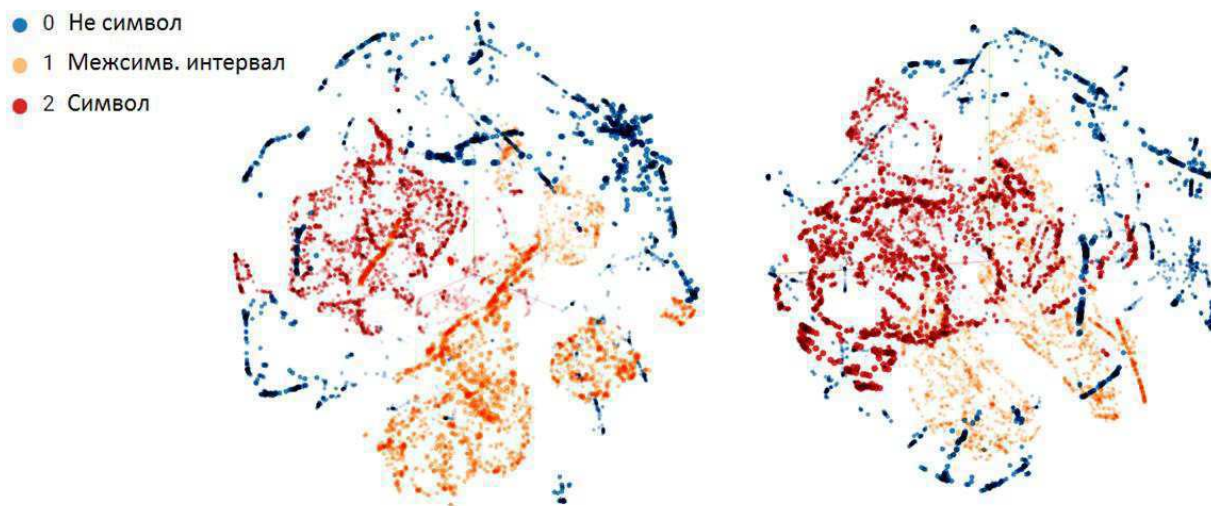


Рисунок 18 – Визуализация тренировочной выборки для детектирования и сегментации с помощью алгоритма t-SNE в трехмерном пространстве (две проекции)

2.3.4 Формирование выборки для распознавания символов

Данная выборка состоит из изображений, входящих в состав положительной части выборки, используемой для детектирования текста, однако все символы отнесены не к одному и тому же классу “Символ”, а к различным, соответствующим данным символам классам определенных цифр и букв.

Для распознавания текста на изображениях, соответствующих выбранной предметной области, необходимо рассматривать большое количество классов, что сильно усложняет процесс формирования набора данных для обучения. На данный момент был создан набор данных, который содержит изображения цифр от 0 до 9, а также буквы Р и D, которые встречаются в числовых столбцах отсканированных документов, выбранных в качестве предметной области для распознавания. В дальнейшем данная выборка будет дополнена буквами кириллицы и латиницы, а также знаками препинания.

На рисунках 19, 20 и 21 представлена визуализированная выборка для классификатора, используемого на этапе распознавания.

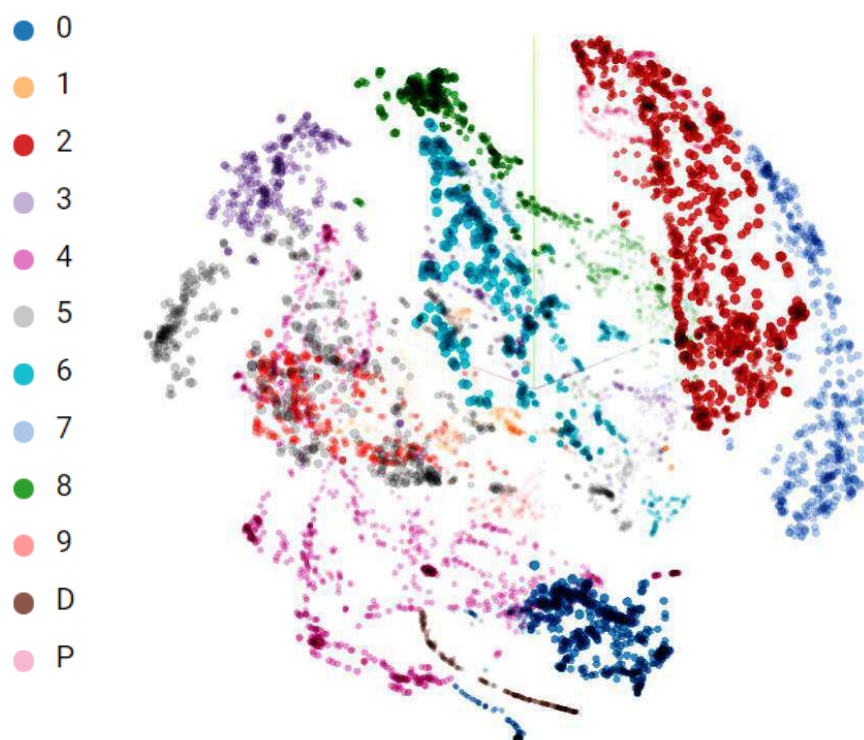


Рисунок 19 – Визуализация тренировочной выборки с помощью алгоритма t-SNE в трехмерном пространстве (проекция 1)

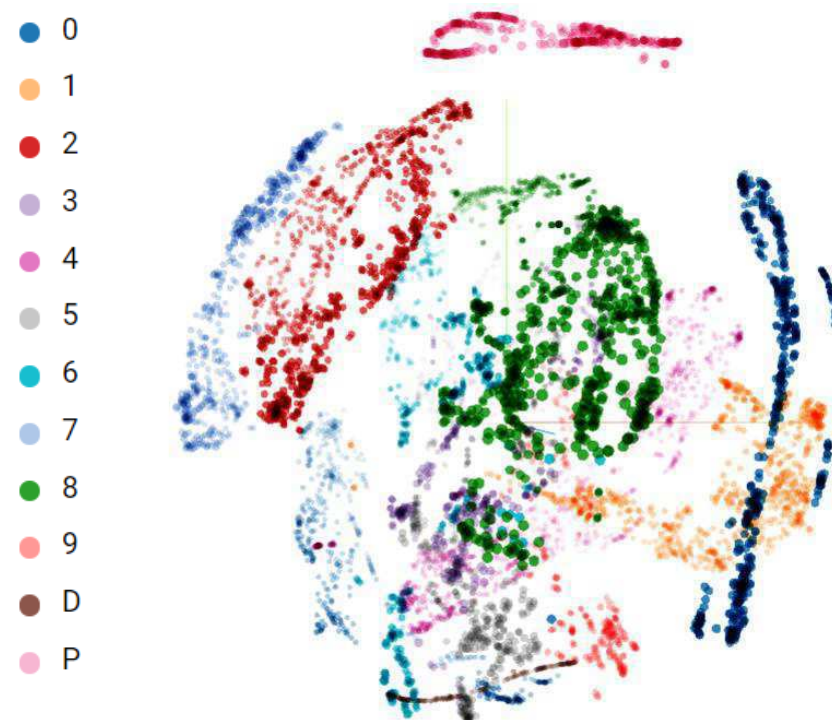


Рисунок 20 – Визуализация тренировочной выборки с помощью алгоритма t-SNE в трехмерном пространстве (проекция 2)

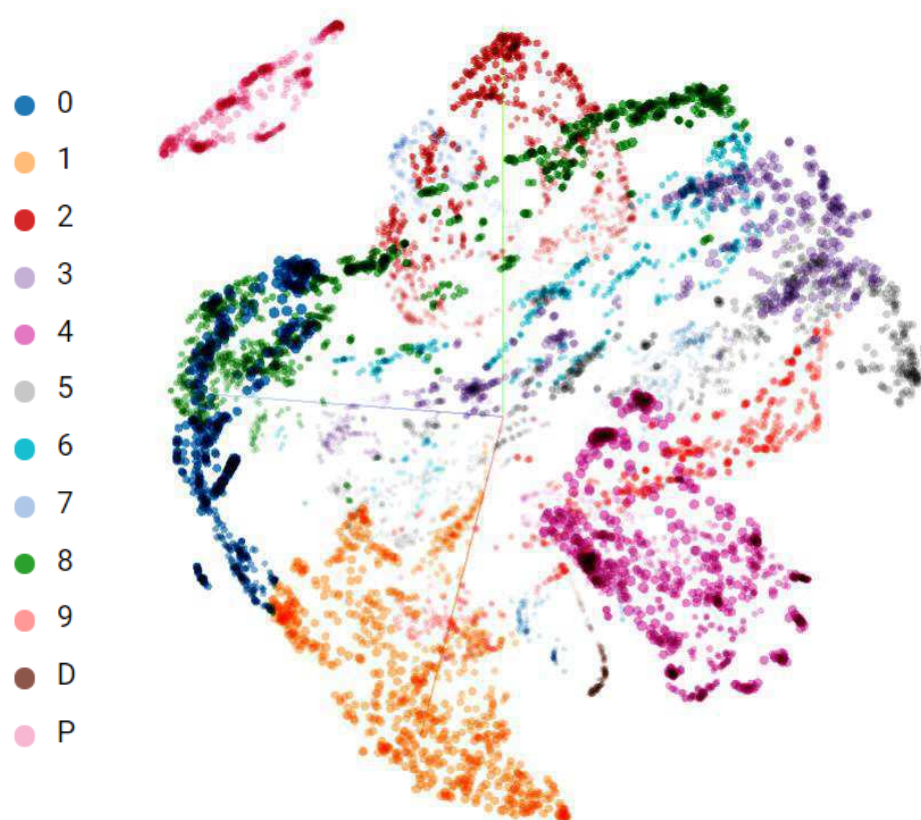


Рисунок 21 – Визуализация тренировочной выборки с помощью алгоритма t-SNE в трехмерном пространстве (проекция 3)

2.3.5 Тестовые выборки

Для проверки спроектированных классификаторов используются тестовые выборки, объекты для которых извлекаются из общего набора данных. Количество объектов, которые используются для обучения нейронных сетей, составляют 70% от общего числа объектов общего набора данных, а объекты, которые формируют тестовую выборку, соответственно 30% от общего набора данных. Объекты, содержащиеся в тестовой выборке, не содержатся в тренировочной выборке.

2.4 API для проектирования нейронных сетей

На данный момент существует несколько библиотек для машинного обучения и нейронных сетей. Рассмотрим некоторые из них.

Tensorflow – свободная программная библиотека для машинного обучения от Google [11]. В данный момент она используется в научно-исследовательской работе и в десятках коммерческих продуктов Google, в том числе в Google Search, Gmail, Photos, Youtube, Translate, Assistant, а также всевозможных системах распознавания, в том числе распознавания речи. Основная библиотека Tensorflow подходит для широкого семейства техник машинного обучения, в том числе и для глубинного обучения.

В дополнение к основной функциональности машинного обучения, TensorFlow также включает собственную систему логирования, а также собственный интерактивный визуализатор логов.

Torch – библиотека для научных вычислений с широкой поддержкой алгоритмов машинного обучения. Разрабатывается Idiap Research Institute, New York University и NEC Laboratories America, начиная с 2000г., распространяется под лицензией BSD [11].

Библиотека реализована на языке Lua с использованием C и CUDA. Быстрый скриптовый язык Lua в совокупности с технологиями SSE, OpenMP, CUDA позволяют Torch показывать неплохую скорость по сравнению с другими библиотеками. На данный момент поддерживаются операционные системы Linux, FreeBSD, Mac OS X. Основные модули также работают и на Windows. В зависимостях Torch находятся пакеты imagemagick, gnuplot, nodejs, npm и другие.

Библиотека состоит из набора модулей, каждый из которых отвечает за различные стадии работы с нейросетями. Так, например, модуль nn обеспечивает конфигурирование нейросети (определению слоев, и их параметров), модуль optim содержит реализации различных методов оптимизации, применяемых для обучения, а gnuplot предоставляет возможность визуализации данных (построение графиков, показ изображений и т.д.). Установка дополнительных модулей позволяет расширить функционал библиотеки. Данная библиотека может быть использована на языках Lua и C.

Torch позволяет создавать следующие слои:

- Полносвязный слой (Linear).
- Функции активации: гиперболический тангенс (Tanh), выбор минимального (Min) или максимального (Max), softmax-функция (SoftMax) и другие.
- Сверточные слои: свертка (Convolution), прореживание (SubSampling), пространственное объединение (MaxPooling, AveragePooling, LPPooling), разностная нормализация (SubtractiveNormalization).

Функции ошибки: средне-квадратичная ошибка (MSE), кросс-энтропия (CrossEntropy) и т.д.

При обучении могут использоваться следующие методы оптимизации:

- Стохастический градиентный спуск (SGD),
- Усредненный стохастический градиентный спуск (Averaged SGD) [9].
- Алгоритм Бройдена – Флетчера – Гольдфарба – Шанно (L-BFGS) [10].
- Метод сопряженных градиентов (Conjugate Gradient, CG).

Microsoft Azure – функциональная среда разработки для задач машинного обучения, поддерживающая язык R (для работы с big data) и Python.

Caffe – библиотека, реализованная с использованием языка программирования C++, имеются обертки на Python и MATLAB [11]. Официально поддерживаемые операционные системы — Linux и OS X, также имеется неофициальный порт на Windows. Caffe использует библиотеку BLAS (ATLAS, Intel MKL, OpenBLAS) для векторных и матричных вычислений. Наряду с этим, в число внешних зависимостей входят glog, gflags, OpenCV, protoBuf, boost, leveldb, nappy, hdf5, lmdb. Для ускорения вычислений Caffe может быть запущена на GPU с использованием базовых возможностей технологии CUDA или библиотеки примитивов глубокого обучения cuDNN.

Pylearn2 — библиотека, разрабатываемая в лаборатории LISA в университете Монреаля с февраля 2011 года [11]. Имеет около 100

разработчиков на GitHub. Библиотека распространяется под лицензией BSD 3-Clause.

Pylearn2 реализована на языке Python, в настоящее время поддерживается операционная система Linux, также возможен запуск на любой операционной системе с использованием виртуальной машины, т.к. разработчики предоставляют сконфигурированную обертку виртуальной среды на базе Vagrant. Pylearn2 является надстройкой над библиотекой Theano. Дополнительно требуются PyYAML, PIL. Для ускорения вычислений Pylearn2 и Theano используют Cuda-convnet, которая реализована на C++/CUDA, что дает значительный прирост в скорости.

В Pylearn2 поддерживается возможность создания полностью связанных и сверточных нейросетей, различных видов автокодировщиков (Contractive Auto-Encoders, Denoising Auto-Encoders) и ограниченных машин Больцмана (Gaussian RBM, the spike-and-slab RBM). Предусмотрены несколько функций ошибки: кросс-энтропия (cross-entropy), логарифмическое правдоподобие (log-likelihood). Имеются следующие методы обучения:

1. Пакетный градиентный спуск (Batch Gradient Descent, BGD).
2. Стохастический градиентный спуск (Stochastic Gradient Descent, SGD).
3. Нелинейный метод сопряженных градиентов (Nonlinear conjugate gradient descent, NCG)

Theano — это расширение языка Python, позволяющее эффективно вычислять математические выражения, содержащие многомерные массивы [11]. Библиотека получила свое название в честь имени жены древнегреческого философа и математика Пифагора — Феано (или Теано). Theano разработана в лаборатории LISA для поддержки быстрой разработки алгоритмов машинного обучения.

Библиотека реализована на языке Python, поддерживается на операционных системах Windows, Linux и Mac OS. В состав Theano входит

компилятор, который переводит математические выражения, написанные на языке Python в эффективный код на C или CUDA.

Theano предоставляет базовый набор инструментов для конфигурации нейросетей и их обучения. Возможна реализация многослойных полностью связанных сетей (Multi-Layer Perceptron), сверточных нейросетей (CNN), рекуррентных нейронных сетей (Recurrent Neural Networks, RNN), автокодировщиков и ограниченных машин Больцмана. Также предусмотрены различные функции активации, в частности, сигмоидальная, softmax-функция, кросс-энтропия. В ходе обучения используется пакетный градиентный спуск (Batch SGD).

В данной работе для реализации нейронных сетей была использована библиотека Tensorflow на языке программирования Python, так как это наиболее динамично развивающаяся библиотека для машинного обучения, поддерживаемая компанией Google. Плюсом также является тот факт, что существует подробная официальная документация, включающая в свой состав множество примеров реализации различных алгоритмов машинного обучения с помощью данного API.

3 Результаты работы программы

3.1 Методика оценки качества работы разработанной системы

Разработанный программный комплекс является составной системой. Так как система состоит из нескольких частей целесообразно оценивать работу как системы в целом, так и ее составных частей. Рассмотрим подробнее методику оценки качества работы различных частей разработанного программного комплекса.

3.1.1 Методы оценки работы алгоритмов классификации

Основой проверки спроектированного классификатора является тестовая выборка, которая содержит множество объектов и поставленные в соответствие им классы. В зависимости от задачи получение подобной выборки может быть затруднено, так как зачастую ее объемы велики, а в ее составлении принимают участие люди, что замедляет процесс ее создания. Иногда этот процесс может быть ускорен заменой ручной работы на программную генерацию тестовых объектов. При проверке работы классификатора необходимо сравнить его решения с заведомо известными верными ответами из тестовой выборки. Для оценки качества его работы могут быть использованы следующие численные критерии:

- 1) точность (Accuracy);
- 2) точность и полнота (Precision и Recall);
- 3) F1-score.

При разработке классификатора в первую очередь интересует обобщающая способность системы. Однако, оценка его точности производится на тех данных, которые принимали участие в процессе обучения. Требуется построить доверительные интервалы для ошибки распознавания с помощью границы Чернова. Граница Чернова позволяет получить вероятность события,

что ошибка распознавания не будет превышать определенный порог γ . Границу Чернова можно вычислить по формуле 13 [12].

$$P(|y - \hat{y}| > \gamma) \leq 2 \cdot \exp(-2\gamma^2 m), \quad (13)$$

где y – вероятность ошибки на новых данных, \hat{y} – вероятность ошибки, оцененная по обучающей выборке, m – размер тренировочной выборки, γ – пороговое значение отклонения вероятности ошибки на новых данных от вероятности ошибки, оцененной по обучающей выборке.

3.1.2 Критерий Accuracy

Критерий Accuracy показывает долю объектов, которые были верно классифицированы в результате работы алгоритма.

$$Accuracy = \frac{P}{N}, \quad (14)$$

где, P – количество объектов, для которых было принято верное решение, а N – размер тестовой выборки.

Недостатком данного метода оценки качества работы классификатора является то, что все решения имеют одинаковый вес, что может быть не корректно если в обучающей выборке в различных классах наблюдается разное количество объектов. В данном случае алгоритм располагает большим количеством информации, описывающей определенные классы, что говорит о более высокой точности при принятии решений в случае этих классов. При оценке качества работы этом может привести к высокому значению метрики, однако классификатор в этом случае может также выдавать и большое количество неверных решений в рамках какого-то одного или нескольких классов.

Решением данного недостатка может оказаться специально подготовленная сбалансированная выборка, которая будет содержать одинаковое количество объектов для каждого класса, но в таком случае алгоритм лишается части информации для обучения, которая может понадобиться для принятия верного решения.

Другой способ решения недостатка критерия Accuracy заключается в изменении подхода к формальной оценке качества с использованием точности и полноты.

3.1.3 Точность и полнота

Точность системы в пределах класса (Precision) – это доля объектов, принадлежащих данному классу, относительно всех документов, которые система отнесла к этому классу.

Полнота системы (Recall) – это доля найденных классификатором объектов, принадлежащих классу, относительно всех объектов этого класса в тестовой выборке.

Эти значения рассчитываются на основании таблицы контингентности, которая составляется для каждого класса отдельно.

Таблица 1 – Пример таблицы контингентности для класса i

		Известное верное решение	
		Положительное	Отрицательное
Решение классификатора	Положительное	TP_i	FP_i
	Отрицательное	FN_i	TN_i

В таблицу контингентности записывается информация о количестве верных и неверных решений системы для объектов данного класса:

- 1) TP_i – количество истинно-положительных решений, принятых для класса i ;
- 2) TN_i – количество истинно-отрицательных решений, принятых для класса i ;
- 3) FP_i – количество ложно-положительных решений, принятых для класса i ;
- 4) FN_i – количество ложно-отрицательных решений, принятых для класса i .

Критерии точность и полнота вычисляются по формулам 15 и 16.

$$Precision(i) = \frac{TP_i}{TP_i + FP_i}, \quad (15)$$

$$Recall(i) = \frac{TP_i}{TP_i + FN_i}. \quad (16)$$

3.1.4 Матрица неточностей

Значения точности и полноты можно рассчитать с использованием матрицы неточностей (confusion matrix). Если количество классов невелико, то составление матрицы неточностей позволит наглядно представить результат работы алгоритма.

Матрица неточностей – это матрица размера N на N , где N – количество классов. Столбцы этой матрицы соответствуют заведомо известным верным решениям, а строки соответствуют решениям классификатора. Матрица инициализируется нулевыми значениями. Для заполнения данной матрицы требуется при классификации объекта тестовой выборки инкрементировать число, находящееся на пересечении строки класса, соответствующего решению классификатора, относительно данного объекта, и столбца, соответствующего заведомо известному классу, к которому действительно принадлежит объект. Элементы главной диагонали будут содержать число верно принятых решений классификатора.

Используя матрицу неточностей A можно вычислить значения критериев точности и полноты по формулам 17 и 18.

$$Precision(i) = \frac{A_{i,i}}{\sum_{j=1}^N A_{i,j}}, \quad (17)$$

$$Recall(i) = \frac{A_{i,i}}{\sum_{j=1}^N A_{j,i}}. \quad (18)$$

3.1.5 F-score

На практике невозможно достигнуть максимальных значений критериев precision и recall одновременно, поэтому требуется найти некоторый баланс. Критерий F-score объединяет в себе информацию о точности и полноте классификатора, представляя собой гармоническое среднее между precision и

recall, что позволяет упростить процесс оценки качества работы алгоритма. Данная метрика стремится к нулю, если точность или полнота стремится к нулю.

Получить численное значение F-score можно по формуле:

$$F(i) = (\beta^2 + 1) \frac{Precision(i) \cdot Recall(i)}{Precision(i) + Recall(i)}, \quad (19)$$

где i – номер класса, β – параметр, отвечающий за приоритет точности или полноты при оценке метрики.

Данная формула позволяет рассчитать F-score, придав различный вес метрикам точности и полноты, если при оценке алгоритма придается приоритет одной из них. Если при оценке отдается приоритет точности, то параметр β должен принимать значения в диапазоне $0 < \beta < 1$, если отдается приоритет полноте то параметр β принимает значения $\beta > 1$. При $\beta = 1$ точность и полнота имеют одинаковый вес, а такой критерий называется сбалансированной или F1-score.

Для оценки качества работы спроектированных классификаторов используется метрика F1-score, так как она позволяет свести к одному числовому значению две основополагающих метрики (точность и полнота), что упрощает принятие решения о том, на сколько положительно влияют изменения, вносимые в алгоритм работы классификатора, на результат его работы.

3.1.6 Оценка качества распознавания текстовой информации

Описанные ранее критерии оценки эффективности относились к задаче анализа алгоритмов машинного обучения. Для конкретных прикладных задач, например, задачи распознавания символов, возможно использование дополнительных критериев, например, расстояния Левенштейна.

Расстояние Левенштейна (редакционное расстояние) – это минимальное количество операций редактирования одного символа (вставки, удаления, замены одного символа на другой), необходимых для превращения одной строки в другую.

Недостатком такого метода сравнения строк является то, что расстояния, полученные для абсолютно разных коротких слов, оказываются небольшими, однако расстояния, полученные для похожих длинных слов, оказываются велики.

При вычислении расстояния Левенштейна веса операций могут быть различными, в зависимости от операции (вставки, удаления или замены). В данной работе при вычислении расстояния Левенштейна веса каждой операции равны.

Итоговая точность алгоритма сильно зависит от эффективной работы алгоритмов на этапах, предшествующих этапу распознавания найденных символов, так как она складывается не только из ошибок распознавания классов букв и цифр, но еще и из ошибочно обнаруженных областей, которые должны содержать текст. Также качество распознанного текста может сильно упасть из-за неправильно сегментированных символов, так как корректная расстановка границ между символами важна при формировании изображений символов, которые будут переданы на этап распознавание. Таким образом небольшая ошибка на ранних этапах может сильно повлиять на качество работы системы в целом.

3.2 Оценка качества работы

3.2.1 Оценка качества работы классификатора, используемого на этапах детектирования и сегментации текста

Данный классификатор принимает решения о принадлежности объекта к одному из следующих классов:

- класс 0 – на изображении нет символа или межсимвольного интервала;
- класс 1 – на изображении представлен межсимвольный интервал;
- класс 2 – на изображении представлен символ.

Таблица 2 – Оценка по метрике Assurasy, полученная на тренировочной выборке

Количество объектов	Верных решений	Assurasy	Доверительный интервал для обобщающей способности (уровень доверия 99%)
27000	26965	0.9987	0.0034

Таблица 3 – Матрица неточностей, полученная на тренировочной выборке

<div>Верное решение</div> <div>Решение классификатора</div>	0	1	2
0	9629	2	0
1	21	9667	0
2	12	0	7669
Всего объектов	9662	9669	7669

Таблица 4 – Оценка по метрикам Precision и Recall, полученная на тренировочной выборке

<div>Метрика</div> <div>Класс</div>	Precision	Recall
0	0.996584	0.999792
1	0.999793	0.997832
2	1.000000	0.998437

Таблица 5 – Оценка по метрике F1-score, полученная на тренировочной выборке

Класс \ Метрика	F1-score
0	0.998185
1	0.998811
2	0.999218

Таблица 6 – Оценка по метрике Accuracy, полученная на тестовой выборке

Количество объектов	Верных решений	Accuracy
11223	11150	0.9935

Таблица 7 – Матрица неточностей, полученная на тестовой выборке

Решение классификатора \ Верное решение	0	1	2
0	3931	13	5
1	30	3964	0
2	24	1	3255
Всего объектов	3985	3978	3260

Таблица 8 – Оценка по метрикам Precision и Recall, полученная на тестовой выборке

Метрика Класс	Precision	Recall
0	0.986449	0.995442
1	0.996481	0.992489
2	0.998466	0.992378

Таблица 9 – Оценка по метрике F1-score, полученная на тестовой выборке

Метрика Класс	F1-score
0	0.990925
1	0.994481
2	0.995413

3.2.2 Оценка качества работы детектора текста

Для оценки работы детектора текста сравниваются маска областей с текстом на изображении, полученная в результате работы детектора текста, с некоторой идеальной маской, полученной в результате выделения текста человеком. Такой метод оценки учитывает влияние не только классификатора на детектирование текста, а еще и способы обработки полученных данных в результате обработки изображения методом скользящего окна с использованием классификатора.

Таблица 10 – Оценка по метрике Accuracy

Количество пикселей	Пикселей совпадает	Accuracy
246304	164650	0.6685

Таблица 11 – Матрица неточностей, полученная на тренировочной выборке

Решение классификатора \ Верное решение	Пиксель, принадлежащий области, содержащей текст	Пиксель, принадлежащий области, не содержащей текст
Пиксель, принадлежащий области, содержащей текст	164640	0
Пиксель, принадлежащий области, не содержащей текст	45027	36637

Таблица 12 – Оценка по метрикам Precision и Recall, полученная на тренировочной выборке

Класс \ Метрика	Precision	Recall
Пиксель, принадлежащий области, содержащей текст	1.000000	0.785245
Пиксель, принадлежащий области, не содержащей текст	0.448631	1.000000

Таблица 13 – Оценка по метрике F1-score, полученная на тренировочной выборке

Класс \ Метрика	F1-score
Пиксель, принадлежащий области, содержащей текст	0.879706
Пиксель, принадлежащий области, не содержащей текст	0.619386

3.2.3 Оценка качества работы классификатора, используемого на этапе распознавания символов

Данный классификатор принимает решения о принадлежности объекта к одному из следующих классов:

- класс 0 – на изображении присутствует цифра 0;
- класс 1 – на изображении присутствует цифра 1;
- класс 2 – на изображении присутствует цифра 2;
- класс 3 – на изображении присутствует цифра 3;
- класс 4 – на изображении присутствует цифра 4;
- класс 5 – на изображении присутствует цифра 5;
- класс 6 – на изображении присутствует цифра 6;
- класс 7 – на изображении присутствует цифра 7;
- класс 8 – на изображении присутствует цифра 8;
- класс 9 – на изображении присутствует цифра 9;
- класс 10 – на изображении присутствует буква D;
- класс 11 – на изображении присутствует буква P.

Таблица 14 – Оценка по метрике Ассигасу, полученная на тренировочной выборке

Количество объектов	Верных решений	Accuracy	Доверительный интервал для обобщающей способности (уровень доверия 99%)
34000	34000	1	0.0034

Таблица 15 – Матрица неточностей, полученная на тренировочной выборке

<div>Верное решение</div> <div>Решение классификатора</div>	0	1	2	3	4	5	6	7	8	9	10	11
0	2840	0	0	0	0	0	0	0	0	0	0	0
1	0	2889	0	0	0	0	0	0	0	0	0	0
2	0	0	2901	0	0	0	0	0	0	0	0	0
3	0	0	0	2856	0	0	0	0	0	0	0	0
4	0	0	0	0	2856	0	0	0	0	0	0	0
5	0	0	0	0	0	2816	0	0	0	0	0	0
6	0	0	0	0	0	0	2867	0	0	0	0	0
7	0	0	0	0	0	0	0	2827	0	0	0	0
8	0	0	0	0	0	0	0	0	2819	0	0	0
9	0	0	0	0	0	0	0	0	0	2843	0	0
10	0	0	0	0	0	0	0	0	0	0	2596	0
11	0	0	0	0	0	0	0	0	0	0	0	2890
Всего объектов	2840	2889	2901	2856	2856	2816	2867	2827	2819	2843	2596	2890

Таблица 16 – Оценка по метрикам Precision и Recall, полученная на тренировочной выборке

Метрика \ Класс	Precision	Recall
0	1.0	1.0
1	1.0	1.0
2	1.0	1.0
3	1.0	1.0
4	1.0	1.0
5	1.0	1.0
6	1.0	1.0
7	1.0	1.0
8	1.0	1.0
9	1.0	1.0
10	1.0	1.0
11	1.0	1.0

Таблица 17 – Оценка по метрике F1-score, полученная на тренировочной выборке

Метрика \ Класс	F1-score
0	1.0
1	1.0
2	1.0
3	1.0
4	1.0
5	1.0
6	1.0
7	1.0
8	1.0
9	1.0
10	1.0
11	1.0

Таблица 18 – Оценка по метрике Ассигасу, полученная на тестовой выборке

Количество объектов	Верных решений	Ассигасу
13600	13596	0.9997

Таблица 19 – Матрица неточностей, полученная на тестовой выборке

<div>Верное решение</div> <div>Решение классификатора</div>	0	1	2	3	4	5	6	7	8	9	10	11
0	1600	0	0	0	0	0	0	1	0	0	0	0
1	0	1110	0	0	0	0	0	0	0	0	0	0
2	0	0	1099	0	0	0	0	1	0	0	0	0
3	0	0	0	1143	0	0	0	0	0	0	0	0
4	0	1	0	0	1144	0	0	0	0	0	0	0
5	0	0	0	0	0	1184	0	0	0	0	0	0
6	0	0	0	0	0	0	1133	0	0	0	0	0
7	0	0	0	0	0	0	0	1171	0	0	0	0
8	0	0	0	0	0	0	0	0	1181	0	0	0
9	0	0	0	0	0	0	0	0	0	1157	0	0
10	0	0	0	0	0	0	0	0	0	0	1004	0
11	0	0	0	1	0	0	0	0	0	0	0	1110
Всего объектов	1160	1111	1099	1144	1144	1184	1133	1173	1181	1157	1004	1110

Таблица 20 – Оценка по метрикам Precision и Recall, полученная на тестовой выборке

Метрика Класс	Precision	Recall
0	1.000000	0.999139
1	0.999010	1.000000
2	1.000000	0.999090
3	0.999126	1.000000
4	1.000000	0.999127
5	1.000000	1.000000
6	1.000000	1.000000
7	1.000000	1.000000
8	0.998250	1.000000
9	1.000000	1.000000
10	1.000000	1.000000
11	1.000000	0.999100

Таблица 21 – Оценка по метрике F1-score, полученная на тестовой выборке

Метрика Класс	F1-score
0	0.999560
1	0.999550
2	0.999545
3	0.999563
4	0.999563
5	1.000000
6	1.000000
7	0.999147
8	1.000000
9	1.000000
10	1.000000
11	0.999550

3.2.4 Оценка качества распознавания

Для оценки качества распознавания разработанной системы используется изображение, представленное на рисунке 22.

82050661	1	P043	0,01	358,80
82050662	1	P043	0,01	478,40
82050663	1	P043	0,01	496,13
82050664	1	P043	0,01	520,05
82050665	8	P043	0,08	2321,12
82050666	1	P043	0,01	713,95
82050667	1	P043	0,01	477,00
82050668	1	P043	0,01	281,70

Рисунок 22 – Исходное изображение

На данном изображении содержится следующий текст:

82050661 1 P043 001 35880
82050662 1 P043 001 47840
82050663 1 P043 001 49613
82050664 1 P043 001 52005
82050665 8 P043 008 232112
82050666 1 P043 001 71395
82050667 1 P043 001 47700
82050668 1 P043 001 28170

Для оценки качества работы алгоритма распознавания текстовой информации вычисляется расстояние Левенштейна между данным текстом, который принимается за эталонный, и текстом, который получается в результате распознавания на выходе системы. В таблице 22 приведены результаты распознавания текста на изображении разработанной системой и программными комплексами ABBYY FineReader 14 и CuneiForm, а также производится сравнение полученных результатов.

Таблица 22 – Сравнение результатов распознавания разработанной системы с результатами других программных комплексов

Система оптического распознавания текста	Текст	Расстояние Левенштейна
Система, разработанная в данной работе	82050641 2 P543 021 338305 42050662 7 P043 01 47840 82050663 7 P043 821 496330 32050664 0 P043 5211 32205 82056665 P043 7 823 232112 82050666 7 P543 5212 71333 82050667 7 P043 821 47700 82050669 7 P943 831 13170	46
ABBYY FineReader	82050061 i ----- PC-43 0,CI j 3S3.Sc "82050662 « i 1 j pw3 i C.Cii £7 S. «1C i 82050663 PC-43 u.ui *436:13} 82050664 7" i ?G*13 и,olj 52C.C5» I 82050665 s PC-43 0,0-8» — 2 321,12! 82050666 i PO-43 11 713.35 \ 82050667 i PO-43 0,011 «477.CC • 82050668 1 PCm13 0,011 2 31.70 j'	131
Cuneiform	82050662 86~3 '82050662 82050663 82050664 v ~. ~28,4. 82050665 82050666 82050667 82050668 8v43 ec 083 03 ,0 8 70'	124

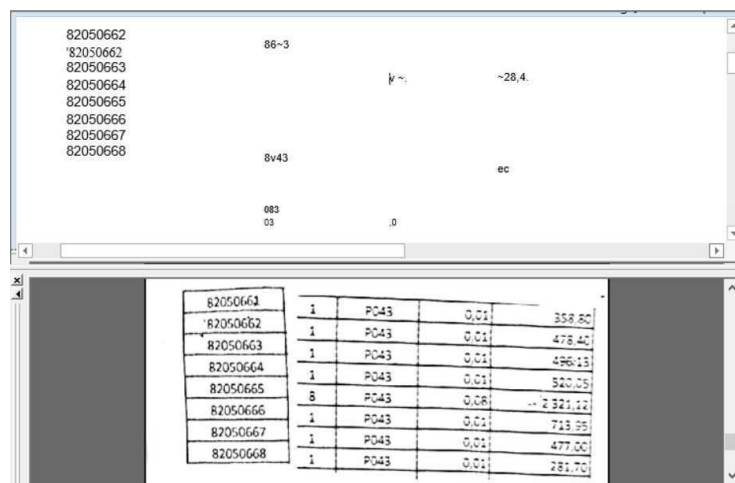


Рисунок 23 – Результат работы программы Cuneiform при решении задачи распознавания текста на заданном изображении

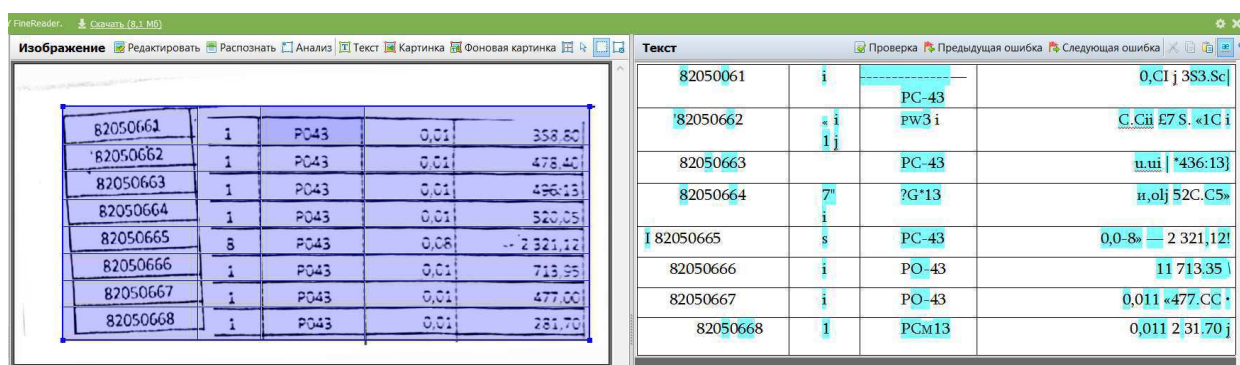


Рисунок 24 – Результат работы программы ABBYY FineReader при решении задачи распознавания текста на заданном изображении

Можно заметить, что разработанная в данной работе система показывает лучшие результаты на данной задаче распознавания, чем другие программные комплексы.

3.3 Исследование влияния функциональных частей системы на результат

Для выявления недостатков разработанной системы производится оценка качества распознавания с заменой некоторых функциональных модулей на идеальные.

Так как при оценке классификатора, используемого на этапе распознавания символов, на тестовой выборке были получены высокие оценки для каждого класса, то замене подвергаются маска областей с текстом,

получаемая на выходе детектора текста, а также маска межсимвольных интервалов, получаемая на выходе этапа сегментации.

Для оценки качества распознавания разработанной системы используется изображение, представленное на рисунке 25.

В качестве результатов работы идеальной системы используются маски, составленные человеком в графическом редакторе, представленные на рисунке 26 и рисунке 27.

82050661	1	P043	0,01	358,80
82050662	1	P043	0,01	478,40
82050663	1	P043	0,01	496,13
82050664	1	P043	0,01	520,05
82050665	8	P043	0,06	2 321,12
82050666	1	P043	0,01	713,95
82050667	1	P043	0,01	477,00
82050668	1	P043	0,01	281,70

Рисунок 25 – Исходное изображение

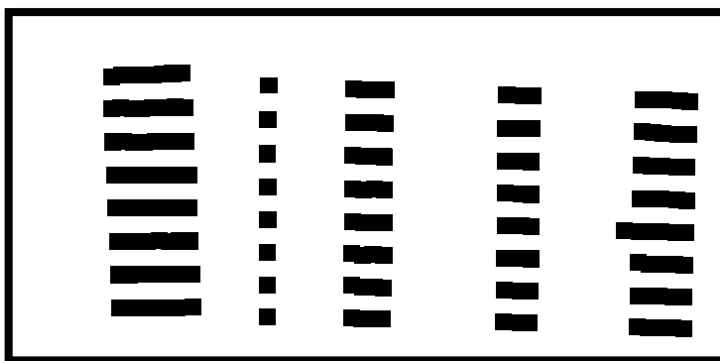


Рисунок 26 – Идеальная маска текстовых областей

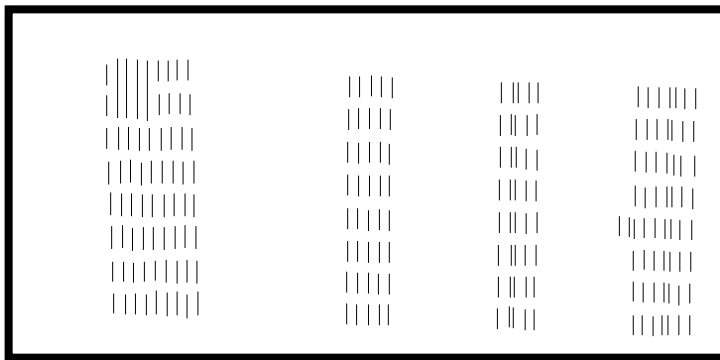


Рисунок 27 – Идеальная маска межсимвольных интервалов

Таблица 23 – Сравнение результатов распознавания разработанной системы

Система оптического распознавания текста	Текст	Расстояние Левенштейна
Разработанная система	82050641 2 P543 021 338305 42050662 7 P043 01 47840 82050663 7 P043 821 496330 32050664 0 P043 5211 32205 82056665 P043 7 823 232112 82050666 7 P543 5212 71333 82050667 7 P043 821 47700 82050669 7 P943 831 13170	46
Разработанная система с использованием идеальной маски текстовых областей и межсимвольных интервалов	82050661 1 P243 821 334030 82050662 1 P043 881 47340 82050663 1 P043 821 496013 82050664 1 P043 001 32225 82056665 8 P043 828 2321012 82050666 1 P543 821 713639 82050667 1 P543 801 47700 82050668 1 P543 801 231672	33
Разработанная система с использованием идеальной маски межсимвольных интервалов	82050661 1 P243 821 333030 82050662 1 P043 821 47340 82050663 1 P043 821 496613 82050664 1 P043 821 32225 82056665 P043 8 828 1321612 82050666 1 P543 821 713633 82050667 1 P043 821 47700 82050669 1 P943 821 231670	41
Разработанная система с использованием идеальной маски текстовых областей	32050641 4243 021 33830 42050662 0 4043 801 47840 82050663 7 P043 821 49613 32050664 P043 021 32205 82056665 8 P243 823 832112 32050666 7 P543 5210 71335 32050667 7 P043 821 47720 82050669 7 P943 831 13170	45

При сравнении результатов можно сказать, что наибольшее негативное влияние на качество распознанного оказывает этап сегментации. Однако при использовании идеальной маски текстовых областей и идеальной маски

межсимвольных интервалов видно, что, несмотря на высокую точность распознавания классификатором символов, возникает много ошибок распознавания. Это говорит о том, что на результат влияет не только качество работы этапов детектирования текста и его сегментации, но еще и способ обработки масок, с помощью которого формируются изображения каждого символа отдельно.

На основании этих выводов можно сказать, что для улучшения качества распознавания системой имеет смысл в первую очередь улучшить качество работы этапа сегментации и алгоритмов обработки масок текстовых областей и межсимвольных интервалов.

Заключение

В результате работы был спроектирован и реализован программный комплекс для распознавания текстовой информации на изображениях, была оценена ошибка при распознавании текста с помощью разработанной системы. Исследована зависимость ошибки распознавания от параметров системы.

В ходе работы произведено сравнение качества распознавания между разработанным и имеющимися программными комплексами. Разработанная система распознавания текста показывает на рассмотренной задаче распознавания результаты лучше, чем ABBYY FineReader 14 и CuneiForm. Для одного и того же распознаваемого изображения расстояние Левенштейна по результату работы системы – 46, в то время как расстояние Левенштейна по результату работы ABBYY FineReader – 131, CuneiForm – 124.

Необходимо отметить, что несмотря на то, что разработанный программный комплекс для рассмотренной выборки демонстрирует точность распознавания выше, чем у коммерческой программы, возможно дополнительное улучшение качества работы спроектированной системы за счет улучшения алгоритмов обработки результатов детектирования и сегментации текста, а также за счет увеличения точности работы алгоритма сегментации текста.

Для возможности автоматического форматирования распознанного текста будут разработаны алгоритмы анализа структуры текста на изображении, а именно алгоритмы обнаружения таблиц, абзацев и т.п.

Список литературы

- [1] М. Л. Моисеевич, Математические методы распознавания образов, МГУ, ВМиК, кафедра "Математические методы прогнозирования", 2002-2004.
- [2] G. i. Ray Smith, «An Overview of the Tesseract OCR Engine,» в *Ninth International Conference on Document Analysis and Recognition*, 2007.
- [3] ABBYY, «ABBYY FineReader 14,» 2017. [Электронный ресурс]. URL: <https://www.abbyy.com/ru-ru/finereader>.
- [4] С. Подлесный, «Команда Джеффри Хинтона победила в конкурсе компьютерного зрения ImageNet с двукратным преимуществом,» 14 Июнь 2013. [Электронный ресурс]. URL: <https://habrahabr.ru/post/183380/>.
- [5] M. Nielsen, «Using neural nets to recognize handwritten digits,» Май 2017. [Электронный ресурс]. URL: <http://neuralnetworksanddeeplearning.com>.
- [6] М. Ю. Сиротенко, «Применение нейросетей в распознавании изображений,» 5 Ноябрь 2009. [Электронный ресурс]. URL: <https://geektimes.ru/post/74326/>.
- [7] Y. LeCun, «Efficient backprop,» *Neural networks: Tricks of the trade. Springer Berlin Heidelberg*, pp. 9-48, 2012.
- [8] N. Otsu, «A threshold selection method from gray-level histograms,» *IEEE Trans. Sys., Man., Cyber*, № 9, pp. 62-66, 1979.
- [9] И. Л. Кафтанников и А. В. Парасич, «Проблемы формирования обучающей выборки в задачах машинного обучения,» *Вестник ЮУрГУ, серия "Компьютерные технологии, управление, радиоэлектроника"*, т. 16, pp. 15-24, 2016.
- [10] НОУ Интуит, «Лекция 12: Нейронные сети. Самоорганизующиеся карты Кохонена,» 2017. [Электронный ресурс]. URL: <http://www.intuit.ru/studies/courses/6/6/lecture/180?page=2>.

- [11] Intel, «Сравнение библиотек глубокого обучения на примере задачи классификации рукописных цифр,» 7 Апрель 2015. [Электронный ресурс]. URL: <https://habrahabr.ru/company/intel/blog/254747/>.
- [12] Andrew NG, «CS229 Lecture notes, Part VI».

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
«Санкт-Петербургский государственный университет
аэрокосмического приборостроения»

ОТЗЫВ РУКОВОДИТЕЛЯ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

на тему Программный комплекс для распознавания текстовой информации
на изображениях

выполненную студентом группы № 5321

Аверьяновым Сергеем Сергеевичем

фамилия, имя, отчество студента

по направлению подготовки/ 11.03.02 Инфокоммуникационные технологии и
специальности (код) (наименование направления подготовки/специальности)

системы связи

(наименование направления подготовки/специальности)

Актуальность темы работы:

Тема бакалаврской работы Сергея Сергеевича связана с задачей оптического распознавания символов на изображениях. Перед студентом была поставлена задача распознавания сканов документов о поставках товаров. Актуальность темы обусловлена современными требованиями к высокой эффективности систем документооборота, которая достигается за счет перехода с бумажных носителей на электронные. При этом существующие программные средства распознавания не всегда предоставляют необходимую точность, в особенности, если сканирование выполнено с искажениями.

Цель и задачи работы:

Целью работы является разработка специализированного решения для распознавания символов на сканированных изображениях документов. Для достижения данной цели перед студентом были поставлены следующие задачи: провести исследование современных методов оптического распознавания символов, проанализировать целевой класс изображений, предложить собственный алгоритм оптического распознавания символов, реализовать предложенный алгоритм и выполнить анализ его эффективности.

Общая оценка выполнения поставленной перед студентом задачи, основные достоинства и недостатки работы:

Все поставленные задачи успешно выполнены. К достоинствам работы необходимо отнести высокую степень проработки процесса формирования обучающей выборки, что является одним из наиболее важных шагов при применении алгоритмов машинного обучения. Также несомненным достоинством является обоснованность и высокая степень проработанности предложенных автором технических решений, позволившая ему повысить точность распознавания символов. В качестве недостатков можно отметить некоторую узкость разработанного решения, которое нацелено на обработку сканов текста, используемых на одном предприятии. Неясно, насколько универсальными являются предложенные Сергеем Сергеевичем концепции и возможно ли адаптировать их для новой прикладной задачи.

Степень самостоятельности и способности к исследовательской работе студента (умение и навыки поиска, обобщения, анализа материала и формулирования выводов):

Все результаты и выводы получены студентом самостоятельно. При работе над заданием студент проявил хорошую теоретическую подготовку, умение применять теоретические знания для решения как исследовательских, так и прикладных задач.

Проверка текста выпускной квалификационной работы с использованием системы "Антиплагиат.ВУЗ" (guap.antiplagiat.ru), проводившаяся «06» июня 2017 г. показывает оригинальность содержания на уровне 87,51%.

Степень грамотности изложения и оформления материала:

Пояснительная записка соответствует требованиям «Положения о выпускной квалификационной работе студентов ГУАП, обучающихся по образовательным программам высшего образования – программам бакалавриата, программам специалитета и программам магистратуры».

Оценка деятельности студента в период подготовки выпускной квалификационной работы (добросовестность, работоспособность, ответственность, аккуратность и т.п.):

Необходимо отметить, что в процессе выполнения квалификационной работы Сергей Сергеевич показал себя как ответственный и добросовестный специалист, способный с высокой степенью самостоятельности выполнять работу в рамках полного цикла, включающего как стадию исследования и анализа, так и стадию разработки программного обеспечения.

Общий вывод:

Считаю, что работа соответствует требованиям, предъявляемым к выпускной квалификационной работе бакалавра по специальности 11.03.02 “Инфокоммуникационные технологии и системы связи”, а ее автор, Аверьянов Сергей Сергеевич, заслуживает присвоения ему квалификации бакалавр.

Руководитель

доц., к.т.н.

должность, уч. степень, звание



07.06.17

подпись, дата

А.И. Веселов

инициалы, фамилия

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

РЕЦЕНЗИЯ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

на тему Программный комплекс для распознавания текстовой информации
на изображениях

выполненную студентом группы № 5321

Аверьяновым Сергеем Сергеевичем

фамилия, имя, отчество студента

по направлению подготовки/
специальности

11.03.02

(код)

Инфокоммуникационные

(наименование направления подготовки/специальности)

технологии и системы связи

(наименование направления подготовки/специальности)

Актуальность темы исследования:

Актуальность темы выпускной работы Аверьянова С.С. не вызывает сомнений, поскольку она направлена на повышение качества распознавания текстовой информации на изображениях, а значит и на повышение эффективности документооборота в целом, что в настоящее время является чрезвычайно важным.

Краткая характеристика структуры работы и отдельных ее разделов:

Пояснительная записка к выпускной бакалаврской работе состоит из введения, трех разделов и заключения.

Во введении описаны цель и задачи работы.

В первом разделе описаны проблемы, связанные с распознаванием текстовой информации, системы распознавания текста, представлена информация об искусственных нейронных сетях.

Во втором разделе детально рассмотрена и проанализирована система распознавания текстовой информации.

Третий раздел посвящен оценке качества разработанного программного комплекса, методам оценки алгоритмов классификации, оценке качества распознавания текстовой информации и анализу результатов выполненных экспериментальных исследований.

Заключение содержит подведение итогов выполненной работы и возможные направления предстоящей работы в данной области.

Достоинства работы (интересные материалы, положения, выводы, в которых проявились самостоятельность студента, его эрудиция, оригинальность мышления, знание литературы, уровень теоретической подготовки и т.п.):

Аверьянов С.С. в выпускной работе продемонстрировал грамотный подход к решению поставленной задачи, высокий уровень подготовки, а также умение логически

обоснованно излагать материал и использовать накопленные теоретические знания на практике. Автор проявил эрудицию в данной предметной области и умение проводить экспериментальные исследования собственной разработки, сопоставление полученных результатов с известными на сегодняшний день решениями рассматриваемой в работе задачи.

Недостатки работы (по содержанию и по оформлению):

К недостаткам выпускной работы Аверьянова С.С. можно отнести следующее:

- в тексте пояснительной записки имеются отдельные отклонения от требований к оформлению пояснительной записки (некорректно оформлен список использованных источников);

- на рис. 5, как это написано, представлено проявление шума квантования спектральных коэффициентов, хотя корректнее, по-видимому, в этой ситуации говорить не только об указанном виде шума, но и наряду с ним о шуме пространственной дискретизации;

- в п. 2.2.1 к числу предварительных методов обработки, позволяющих улучшить качество изображения для последующего распознавания, следовало бы отнести и использование ряда операций морфологической обработки изображений;

- поскольку, как утверждает автор, в результате работы был спроектирован и реализован программный комплекс, следовало бы привести в приложении коды программ.

Общий вывод о выпускной квалификационной работе, ее соответствии предъявляемым к данному виду работ требованиям:

Поставленные перед Аверьяновым С.С. задачи в ходе выполнения выпускной работы решены в полном объеме. Пояснительная записка к выпускной работе удовлетворяет предъявляемым требованиям. В целом представленная на рецензию выпускная работа производит очень хорошее впечатление.

Мнение рецензента об оценке работы:

Считаю, что бакалаврская работа Сергея Сергеевича Аверьянова заслуживает оценки «отлично».

Рецензент

Доцент каф.53, к.т.н., доцент

должность, уч. степень, звание

 09.06.2017

подпись, дата

О.И.Красильникова

инициалы, фамилия