

Одесский национальный политехнический университет

На правах рукописи

Нгуен Тхи Кхань Тьен

УДК 004.932

**ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ ЛОКАЛИЗАЦИИ ТЕКСТОВЫХ
ОБЛАСТЕЙ НА ИЗОБРАЖЕНИЯХ СО СЛОЖНЫМ ФОНОМ**

Специальность 05.13.06 – Информационные технологии

Диссертация на соискание научной степени
кандидата технических наук

Научный руководитель

Николенко Анатолий Александрович
кандидат технических наук,
доцент

Одесса – 2015

СОДЕРЖАНИЕ

СПИСОК УСЛОВНЫХ ОБОЗНАЧЕНИЙ	6
ВВЕДЕНИЕ	7
РАЗДЕЛ 1	
АНАЛИЗ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ЛОКАЛИЗАЦИИ ТЕКСТОВЫХ ОБЛАСТЕЙ НА ИЗОБРАЖЕНИЯХ СО СЛОЖНЫМ ФОНОМ ПРИ ПЕРЕРАБОТКЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ	13
1.1. Анализ существующих информационных систем переработки изображений с текстовой информацией	13
1.2. Постановка задачи локализации текстовых областей	17
1.3. Анализ существующих методов локализации текстовых областей на изображении	20
1.3.1. Методы обнаружения текстовых областей, основанные на знаниях	27
1.3.2 Методы обнаружения текстовых областей, основанные на инвариантных признаках	28
1.3.3. Методы обнаружения текстовых областей, основанные на сравнении с шаблоном	30
1.4. Особенности применения непрерывного вейвлет-преобразования при локализации текстовых областей	32
1.5. Особенности применения дискретного вейвлет-преобразования при локализации текстовых областей	34
1.6. Особенности использования свёрточных нейронных сетей для локализации текстовых областей	36
1.6.1. Особенности обучения сверточной нейронной сети	40

1.7. Методы оценки качества обнаружения и локализации текстовых областей на изображении 43

1.8. Выводы. Постановка задачи исследований 45

РАЗДЕЛ 2

МОДЕЛИ ПРЕДСТАВЛЕНИЯ И МЕТОДЫ ФОРМИРОВАНИЯ ТЕКСТОВОЙ ОБЛАСТИ НА ИЗОБРАЖЕНИИ В СИСТЕМАХ ПЕРЕРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ 48

2.1. Обобщенная схема информационной технологии локализации текстовых областей на изображениях 48

2.2. Квазипериодическая модель строки изображения, которое содержит символы текста 51

2.3 Исследование квазипериодической последовательности в области вейвлет-преобразований 55

2.4. Вероятностная модель представления символа текста 61

2.4.1. Характеристики области символа каждой группы символов 62

2.4.2. Характеристики областей изображения символов букв и цифр. 65

2.5. Метод формирования текстовой области на основе применения вероятностной модели символов 67

2.6. Выводы по разделу 2 69

РАЗДЕЛ 3

РАЗРАБОТКА И ИССЛЕДОВАНИЕ МЕТОДОВ И ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ ЛОКАЛИЗАЦИИ ТЕКСТОВЫХ ОБЛАСТЕЙ НА ИЗОБРАЖЕНИЯХ СО СЛОЖНЫМ ФОНОМ 71

3.1. Корреляционно-экстремальный метод обнаружения текстовых областей на изображениях 71

3.2. Реализация корреляционно-экстремального метода обнаружения текстовых областей	73
3.2.1. Формирование пирамиды изображений для выделенного фрагмента	73
3.2.2. Разработка структуры свёрточной нейронной сети для локализации текстовых областей на полутоновых изображениях со сложным фоном	76
3.2.3. Разработка структуры свёрточной нейронной сети для локализации текстовых областей на цветных изображениях со сложным фоном	79
3.3. Алгоритм обучения предложенной свёрточной нейронной сети	84
3.3.1. Определение параметров карт слоев свёрточной нейронной сети	85
3.3.2. Определение ошибок в сети	93
3.3.3 Коррекция весов связи между слоями и значений смещения	95
3.4 Исследование применения свёрточной нейронной сети для обнаружения текстовых областей на изображениях со сложным фоном	96
3.5. Метод локализации текстовых областей на изображениях со сложным фоном на основе свёрточной нейронной сети с использованием многомасштабного представления изображения на основе ДВП	99
3.6 Выводы по разделу 3.	101
РАЗДЕЛ 4	
РАЗРАБОТКА И ТЕСТИРОВАНИЕ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ ЛОКАЛИЗАЦИИ ТЕКСТОВЫХ ОБЛАСТЕЙ	
4.1. Обобщенная структура информационной технологии локализации текстовых областей на изображении	103

4.2. Разработка специализированного программного комплекса	109
4.3. Тестирование работы свёрточной нейронной сети при локализации текстовых областей	116
4.4. Выводы по разделу 4.	121
ВЫВОДЫ	123
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	126
Приложение А – Фрагменты программного обеспечения	140
Приложение Б – Документы, подтверждающие внедрение результатов работы	201

СПИСОК УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ТО – текстовая область

ИТ – информационная технология

ИП – информативный пиксель

ФП – фоновый пиксель

ОС – область символа

ИРЭМ – интегральная функция распределения энергии по масштабам вейвлет-преобразования

ВП – вейвлет-преобразование

ВФ – вейвлет-функция

ДВП – дискретное вейвлет-преобразование

НВП – непрерывное вейвлет-преобразование

ГВП – гиперболическое вейвлет-преобразование

ВЧ – высокочастотный фильтр

НЧ – низкочастотный фильтр

OCR – Optical Character Recognition (система оптического распознавания)

СНС – свёрточная нейронная сеть

КЭМО ТО – корреляционно-экстремальный метод обнаружения текстовых областей

УМЛ ТО – усовершенствованный метод локализации текстовых областей

ВВЕДЕНИЕ

Одной из задач, решаемых информационными системами переработки графической информации в условиях постоянного роста ее объемов, является задача распознавания текста на изображениях (фотографиях, видео-файлах, отсканированных документах и т.п.). Эта задача требует разработки и использования современных прикладных информационных технологий (ИТ) распознавания текста для автоматизации документооборота, внедрения паспортно-визовой информации, ведения учета подвижного транспорта, идентификации изделий, поиска текстовой информации в мультимедийных коллекциях и тому подобное.

Одной из базовых составляющих ИТ распознавания текста является процедура локализации текстовых областей (ТО) на изображениях, применение которой позволяет сократить объем обрабатываемой информации и определяет оперативность и качество информационных систем переработки графической информации. В результате выполнения этой процедуры определяются координаты границ ТО. Однако, на практике качество локализации ТО на изображении существенно снижается в зависимости от сложности и неоднородности фона и влияния различных искажающих факторов: неравномерности освещения изображения, различного расположения строк текста по отношению к фото- или видеорегирующей камеры и тому подобное.

Проведенный анализ показал, что при наличии сложного и неоднородного фона на изображениях лучшие результаты при оценке качества локализации ТО принадлежат методам на основе корреляционно-экстремального подхода. Однако, их недостатки – большое количество моделей, большие затраты времени как в процессе разработки моделей, так и в процессе локализации ТО, отсутствие инвариантности к масштабу изображений символов – снижают оперативность систем переработки графической информации, использующих эти методы.

Таким образом, возникает противоречие между качеством локализации ТО и оперативностью этой процедуры, особенно при наличии сложного фона на обрабатываемых изображениях. Поэтому создание информационной технологии локализации текстовых областей на изображениях со сложным фоном на основе новых моделей и методов, что позволит устранить указанное противоречие или уменьшить его негативное влияние, является важной научно-технической задачей, которая решается в данной работе.

Связь работы с научными программами, планами, темами. Работа выполнена в соответствии с приоритетными направлениями исследований Одесского национального политехнического университета (ОНПУ), в соответствии с координационными планами Министерства образования и науки Украины, в частности, в рамках научных исследований по госбюджетным научно-исследовательским работам кафедры информационных систем ОНПУ при непосредственном участии автора: «Анализ и распознавание изображений на базе субградиентных итеративных алгоритмов в пространстве вейвлет-преобразования» (№ гос. регистрации 0109U008468), «Исследование и разработка методов и алгоритмов распознавания визуальной информации и анализа сцен» (№ гос. регистрации 0111U010458).

Цель и задачи исследования. Целью диссертационной работы является повышение качества и оперативности локализации текстовых областей на изображениях со сложным фоном в информационных технологиях переработки графической информации.

Для достижения поставленной цели решены следующие **задачи**:

- проведен анализ информационных технологий локализации текстовых областей на изображениях со сложным фоном, используемых при переработке графической информации;
- создана модель представления символа текста на изображении;

– разработан метод формирования текстовых областей на изображениях путем использования вероятностной модели представления символа текста при формировании обучающей выборки для обучения классификатора;

– разработан метод локализации текстовых областей на изображениях со сложным фоном на основе свёрточной нейронной сети.

– разработана на основе предложенных модели и методов информационная технология локализации текстовых областей на изображениях для автоматической переработки графической информации.

Объект исследования – процесс локализации текстовых областей на изображениях в системах переработки графической информации.

Предмет исследования – модели и методы локализации текстовых областей на изображениях со сложным фоном.

Методы исследования. Теория вероятностей и методы цифровой обработки изображений использованы при создании модели представления символа и метода формирования текстовых областей на изображении; теория искусственных нейронных сетей, методы вейвлет-анализа использованы при разработке корреляционно-экстремального метода выявления текстовых областей и метода локализации текстовых областей на изображениях со сложным фоном на основе свёрточной нейронной сети; методы теории вероятности, математической статистики и имитационного моделирования использованы при разработке и апробации ИТ локализации ТО.

Научная новизна полученных результатов:

– получила дальнейшее развитие вероятностная модель представления символа текста путем учета распределения вероятности интенсивности в двумерной области изображения символа, что позволило формировать изображение псевдосимволов текста с вероятностными характеристиками, которые соответствуют реальным символам;

– впервые разработан метод формирования текстовых областей на изображениях за счет использования вероятностной модели представления

символа текста при формировании обучающей выборки, который позволил автоматизировать процесс создания представительной выборки, что устранило ее субъективность;

– получил дальнейшее развитие корреляционно-экстремальный метод обнаружения текстовых областей на изображениях, реализованный на базе свёрточной нейронной сети, структура которой учитывает многомасштабное представление изображения на основе двумерного дискретного вейвлет-преобразования, что позволило обеспечить высокое качество и оперативность процедуры выявления ТО и уменьшить зависимость результатов выявления ТО от размеров и типов шрифтов текста;

– усовершенствован метод локализации текстовых областей на изображении на основе разработанного корреляционно-экстремального метода выявления ТО и их морфологической обработки, что позволило обеспечить высокое качество и оперативность процедуры локализации текстовых областей на изображениях со сложным фоном в системах переработки графической информации.

Практическое значение полученных результатов заключается в том, что разработанные модели и методы реализованы в информационной технологии локализации текстовых областей на изображениях со сложным фоном. Разработанная информационная технология позволяет повысить эффективность локализации текстовых областей на изображениях со сложным фоном. Результаты тестирования показали, что качество локализации текстовых областей с помощью разработанной ИТ на 10-15 % выше, чем качество локализации ТО такими программными продуктами как FreeOCR (onlineOCR.net) и FineReader11.

Метод локализации текстовых областей на изображениях со сложным фоном на основе свёрточной нейронной сети и информационная технология локализации текстовых областей были использованы для моделирования и разработки программно-алгоритмического обеспечения при автоматизации

учета накопленной графической и видеоинформации в научно-производственном объединении «Дискрет» (г. Одесса), а именно, модификации системы автоматизации проверки качества упаковки продукции и правильности распределения ее при формировании партий товара, что позволило повысить эффективность функционирования склада продукции на предприятии.

Основные положения диссертационного исследования внедрены в учебный процесс Одесского национального политехнического университета на кафедре информационных систем при преподавании дисциплин «Методы и средства компьютерных информационных технологий», «Методы и средства искусственного интеллекта», «Вычислительный интеллект» и при выполнении курсового проектирования.

Личный вклад соискателя. Все научные положения и практические результаты, представленные в диссертационной работе, были получены автором лично [1]. В работах, опубликованных в соавторстве, личный вклад соискателя такой: [2, 3, 4, 5, 6, 7] – проведен аналитический обзор существующих методов, моделей и подходов к решению задачи локализации текстовых областей на изображениях со сложным фоном; [8, 9, 10] – разработан и исследован метод обнаружения и локализации текстовых областей и выделение символов на изображениях с неоднородным фоном [11, 12, 13] – разработаны метод локализации текстовых областей на изображениях со сложным фоном на основе свёрточной сети и структура свёрточной нейронной сети для локализации текстовых областей; [5] – разработана вероятностная модель представления символа текста и метод формирования текстовых областей на основе вероятностной модели представления символа; [6] – разработана информационная технология локализации текстовых областей на изображениях со сложным фоном.

Апробация результатов диссертации. Научные результаты и основные положения диссертационной работы докладывались и обсуждались на научных семинарах кафедры информационных систем ОНПУ и научных конференциях:

международных научно-практических конференциях «Современные информационные и электронные технологии» (СИЕТ-2010, 2012, 2013, 2014, г. Одесса), XI Всеукраинской международной конференции «Обработка сигналов и изображений и распознавания образов» (УкрОБРАЗ'2012, г. Киев), VIII Международной научно-практической конференции «Искусственный интеллект. Интеллектуальные системы – 2012» (п. Кацивели), XIII международной научно-технической конференции «Проблемы информатики и моделирования» (Харьков – Ялта, 2013).

Публикации. По теме диссертации опубликовано 15 работ, в том числе 6 статей в научных журналах из перечня профессиональных изданий Украины (из которых 3 статьи в изданиях, входящих в наукометрические базы), 9 в трудах или тезисах докладов международных и всеукраинских конференций.

РАЗДЕЛ 1

АНАЛИЗ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ЛОКАЛИЗАЦИИ ТЕКСТОВЫХ ОБЛАСТЕЙ НА ИЗОБРАЖЕНИЯХ СО СЛОЖНЫМ ФОНОМ ПРИ ПЕРЕРАБОТКЕ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

1.1. Анализ существующих информационных систем переработки изображений с текстовой информацией

Графическое представление информации (специализированные изображения (медицинские, спутниковые и т.д.), фотографии, отсканированные документы, видео-файлы и т.п.) широко используется в быту и технических системах. В зависимости от функционального назначения систем переработки технической информации различают ряд информационных технологий переработки графической информации, которые предназначены для решения таких задач, как: поиск изображений заданного содержания в базах данных, сортировки изображений по определенным критериям, конвертирования изображений, распознавания объектов с определенными признаками на изображении и т.д. (рис. 1.1).

Как видим, для большого количества практических приложений, таких как системы учета, бухгалтерский документооборот, охранный деятельность, мультимедиа и др. важной задачей переработки графической информации является задача распознавания текста на изображении.

Повышенный интерес к системам автоматического распознавания текстов вызван развитием человеко-машинных интерфейсов. Несмотря на то, что разработка методов распознавания текстов ведется уже несколько десятилетий, эта задача по-прежнему далека от автоматизированного решения для реальных изображений со сложным и неоднородным фоном, при отсутствии четких критериев отличия текста от фона [16 - 22].



Рис. 1.1. Взаимосвязь видов графической информации и задач ее обработки.

Для таких задач различают два типа текста на фото- и видеоизображениях (рис. 1.2):

– сюжетный текст (текст сцены) – текст, который естественным образом возникает во время съемки кадра (надписи на рекламных щитах, плакатах, вывесках, дорожных знаках и т.д.), и, как правило, слабо связанный с изображением;

– искусственный (наложенный) текст – текст, который накладывается человеком на изображение при помощи графических редакторов (например, CorelDRAW, Microsoft Paint и т.п.) с целью описать содержимое изображения или дать дополнительную информацию, связанную с ним.

Разрабатываемые в данной работе модели, методы и информационные технологии предназначены для обнаружения фрагментов сюжетного текста как более сложной задачи из-за разного освещения, контрастности, сложных трансформаций стиля текста, сложного фона.



а

б

Рис. 1.2. Пример фотоизображения с наличием:
а – сюжетного текста; б – искусственного текста.

Применительно к областям с сюжетным текстом *фон считается сложным* в случае, когда текст на изображениях не отделен от другой информации, а является частью этой информации, причем текстовая область может произвольно располагаться по отношению к изображениям присутствующих объектов, в том числе и накладываться на них.

Примерами текстовых областей на сложном фоне являются текстовые надписи на памятниках, номера домов и названия улиц на соответствующих указателях, надписи на рекламных плакатах, надписи на дорожных указателях, информационная текстовая строка на экране телевизора и т.д. (рис. 1.3).

Информационные системы переработки изображений с текстовой информацией условно можно разделить на 2 класса:

- системы оптического распознавания символов (OCR – Optical Character Recognition), предназначенные для перевода текста из графического представления в текстовый формат, пригодный для дальнейшего редактирования.

- системы распознавания текста специализированного назначения, предназначенные для распознавания маркировки транспортных средств, поиска ТО в мультимедийных коллекциях, на плакатах и др.



Рис. 1.3. Примеры изображений, содержащих ТО на сложном фоне:
 а – при распознавании указателей; б – при распознавании номеров автомобилей;
 в – при распознавании текста на телевизионном кадре;
 г – при поиске текстовой информации в рекламных изображениях.

Современные OCR системы (например, FineReader, CuneiForm и др. [23 - 25]) способны достигать наилучшей точности распознавания символов – свыше 99 % для «идеальных» изображений, составленных из стандартных шрифтов. Однако, в реальных практических задачах точность распознавания текста, особенно на изображениях со сложным фоном, не превышает 80%, что недопустимо для большинства приложений. К факторам, усложняющим процесс распознавания текста и существенно снижающим точность распознавания, относится низкое качество локализации ТО на изображениях со сложным и зашумленным фоном;

При построении систем распознавания текста специализированного назначения [26, 27, 29-42] возникают трудности, приводящие к низкому качеству распознавания. Эти трудности связаны с тем, что ТО на изображениях

характеризуются разным положением (не только горизонтальным либо вертикальным), сильно отличающимися шрифтами, сложным фоном, наличием таблиц и др. Преодоление этих трудностей может быть достигнуто только за счет улучшения качества локализации ТО.

Таким образом, в результате анализа недостатков информационных систем переработки изображений с текстовой информацией с одной стороны установлено, что основной проблемой, ограничивающей широкое использование систем распознавания текста на практике, является невысокая достоверность распознавания, особенно на сложном фоне, в условиях помех со значениями отношения сигнал/шум по амплитуде меньше 10, которая обусловлена ошибочностью интерпретации изображений за счет недостаточной точности и помехоустойчивости локализации отдельных ТО.

С другой стороны, информационные системы переработки изображений с текстовой информацией, используемые для распознавания сюжетных текстов на изображениях со сложным фоном, как правило, основаны на двух ИТ, обеспечивающих две основные процедуры: локализацию текстовой области (ТО) и непосредственное распознавание текста. При этом успешное выполнение второй процедуры невозможно без качественных результатов первой. Однако создание ИТ для автоматической локализации текстовых областей является непростой задачей из-за переменчивых условий визуализации, связанных с освещением, разным положением текста по отношению к камере, из-за невозможности заранее предугадать, в какой области изображения расположен текст (особенно при наличии разнородной графической информации) и какое этот текст имеет искажение.

1.2. Постановка задачи локализации текстовых областей

Как отмечено выше, ИТ технология локализации ТО является составной частью ИТ распознавания текста. Основные задачи, решаемые в процессе

распознавания текста, можно в соответствии с приведенной схемой (рис. 1.4) разделить на два типа: обнаружение текста и распознавание текста, причем эффективное решение первой оказывает существенное влияние на качество распознавания текста [43].

В свою очередь, тип задач относящихся к обнаружению текста включает: локализацию текстовых строк на сложном фоне без распознавания отдельных символов, извлечение признаков ТО, верификацию полученных областей. Такое представление задач, решаемых перед выполнением сегментации на отдельные символы и их распознаванием, обеспечивает ряд преимуществ для систем переработки графической информации. Рассмотрим их более подробно.

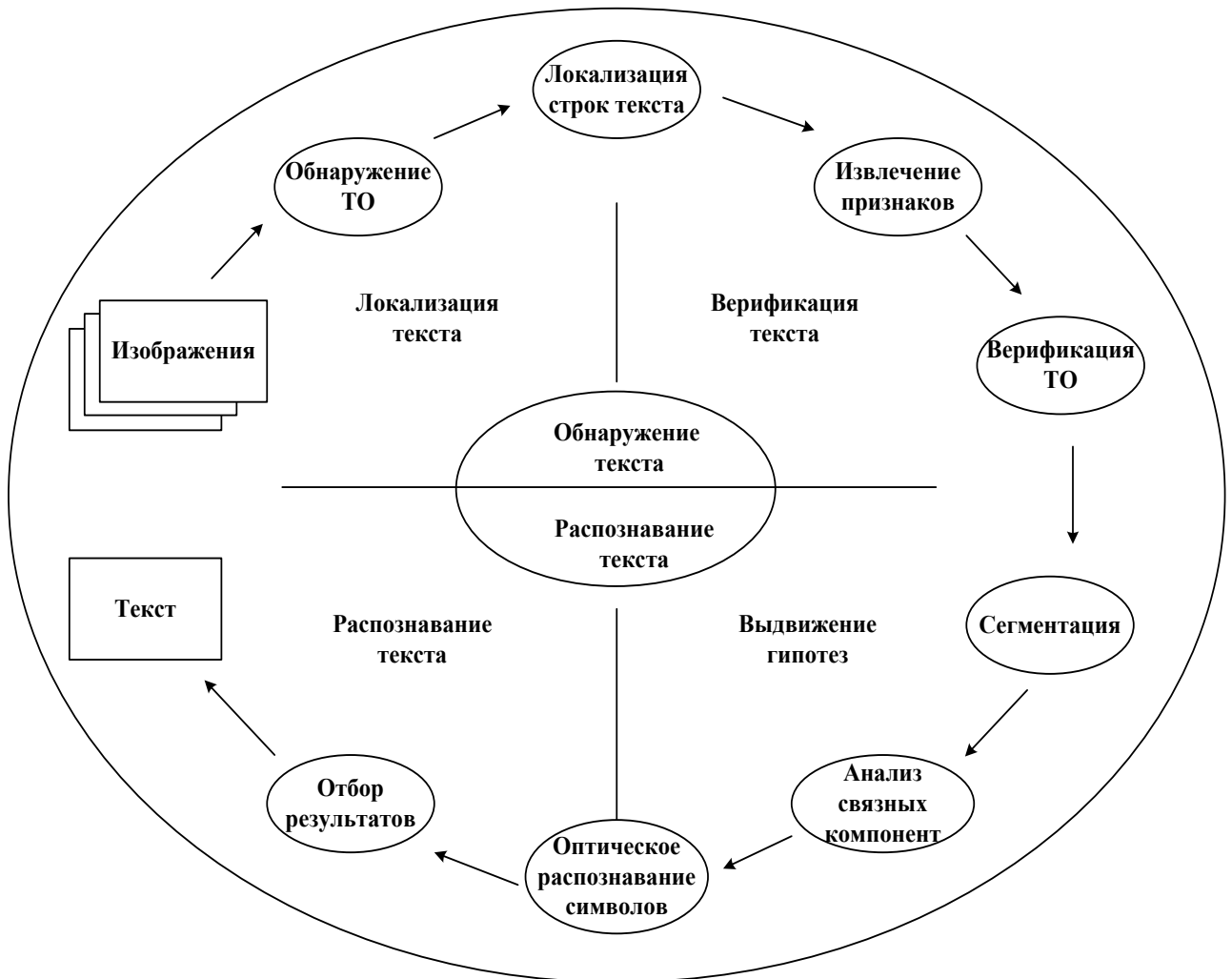


Рис. 1.4. Задачи, решаемые в процессе распознавания текста

Во-первых, встроенный в изображения или видео текст, как правило, не располагается на всём изображении, поэтому не экономично выполнять сегментацию и распознавание символов на участках изображения, не содержащих ТО.

Во-вторых, с помощью точной локализации ТО можно получить дополнительную информацию о масштабах самого текста, что весьма полезно при отделении ТО от фона. Кроме того, фон внутри локализованной ТО, как правило, менее сложен, чем на целом изображении в случае четкой различимости символов текста. Таким образом, хорошее обнаружение ТО обычно способствует лучшей сегментации и распознаванию символов текста.

В-третьих, дополнительные характеристики текстовой строки могут быть использованы для поиска текста. Так как текстовые строки имеют типичные формы и выровнены построчно, то локализация строк текста проще и надежнее, чем локализация отдельных символов.

Таким образом, задача локализации ТО состоит в определении наличия или отсутствия ТО на исходном изображении, и, в случае наличия ТО, нахождения координат и размера каждой из них.

Формально задачу локализации ТО на изображении можно сформулировать следующим образом: для произвольного изображения необходимо найти все прямоугольники, содержащие ТО,

$$FRect_i = \{fx_{li}, fy_{li}, FW_i, FH_i\}, \quad (1.1)$$

где fx_{li}, fy_{li} – координаты верхнего левого угла прямоугольника;

FW_i, FH_i – ширина и высота прямоугольника для i -ой ТО.

Для прямоугольников, ограничивающих ТО, формула (1.1) может быть представлена также в виде

$$FRect_i = \{fx_{li}, fy_{li}, fx_{ri}, fy_{ri}\}, \quad (1.2)$$

где $fx_{li}, fy_{li}, fx_{ri}, fy_{ri}$ – координаты верхнего левого и нижнего правого углов прямоугольника соответственно.

А ширину FW_i и высоту FH_i прямоугольника для i -ой ТО с использованием (1.2) можно определить соответственно как:

$$FW_i = |fy_{li} - fy_{ri}|, FH_i = |fx_{li} - fx_{ri}|.$$

1.3. Анализ существующих методов локализации текстовых областей на изображении

Для решения перечисленных задач распознавания ТО с учетом формализации задачи локализации ТО проведена классификация методов и подходов локализации сюжетных ТО на изображениях со сложным фоном.

По возможностям анализа изображений, среди существующих методов обнаружения и локализации текста можно выделить три подхода: корреляционно-экстремальный, на основе сегментации спектральных текстур, и структурный.

При использовании корреляционно-экстремального подхода для обнаружения ТО для каждой области изображения и набора моделей ТО исчисляются взаимные корреляционные функции, определяется максимальное значение и принимается классификационное решение – «текстовая область» или «не текстовая область» [43, 44]. Преимущества корреляционно-экстремального подхода: хорошая помехоустойчивость в условиях сложного и неоднородного фона, учет пространственного расположения символов в ТО, недостатки – большие вычислительные затраты, большое количество моделей, большие затраты времени как в процессе разработки моделей, так и в процессе локализации, отсутствие инвариантности к масштабу изображения символов.

При подходе на основе сегментации спектральных текстур надпись рассматривается как квазипериодическая последовательность импульсов

[2, 4, 5, 46, 47, 48]. Обнаружение ТО осуществляется с использованием различных фильтров и преобразований: Фурье, Габора, вейвлет-преобразования и т.д. Недостатки – не учитывается пространственное расположение символов, низкая помехоустойчивость в условиях сложного и неоднородного фона.

При структурном подходе обнаруживаются структурные примитивы (буквы, символы) и оценивается порядок их пространственного расположения (т.е. проверяется, образуют ли они надпись) [49, 50, 51]. Предлагается предварительно построить с помощью аппарата математической морфологии [52] или на основе геометрических построений [53] скелетон бинарного изображения символа текста. Затем скелетон превращается в изображение определенного размера, для которого исчисляются признаки, которые и используются при классификации. Также изображение скелетона применяется для решения задачи сегментации изображения слов в символы [51]. Алгоритм состоит в поиске характерных точек на изображении скелетона и контуре текстовых символов, которые затем объединяются по специальному алгоритму в линии, разделяющие изображения слов на сегменты (символы). Недостатки такого подхода – низкая помехоустойчивость в условиях сложного и неоднородного фона.

По порядку поиска ТО, существующие методы обнаружения и локализации текста можно объединить в три группы [43].

1. Восходящие методы: они сегментируют изображения на области и группируют области символов в слова. Применение восходящих методов, в некоторой степени, может избавить от выполнения процедуры обнаружения текста. Из-за трудности в разработке достаточно эффективного алгоритма сегментации для текста на сложном фоне, методы ненадежно работают при обнаружении текста на многих классах изображений и видео.

2. Эвристические нисходящие методы: они сначала обнаруживают ТО на изображениях с использованием эвристических фильтров, а затем используют восходящие методы для распознавания внутри текстовых областей. Эти методы

способны обрабатывать более сложные изображения, чем восходящие методы. Тем не менее, вручную разработанные фильтры являются эмпирическими и, следовательно, обнаруживают множество ложных ТО, которые воспринимаются как ложные тревоги.

3. Нисходящие методы, основанные на обучении: этап обнаружения ТО выполняется с использованием фильтров, обученных с помощью средств машинного обучения.

Среди выделенных групп методы машинного обучения для нисходящих методов локализации ТО имеют наибольший потенциал для обеспечения максимальной производительности. Однако эти методы имеют недостаток в виде высокой стоимости вычислений и трудности создания моделей различных оттенков интенсивности и размеров текстовых строк полутонных изображений при наличии сложного фона.

Интересен метод [43], который относится к категории нисходящих. Этот метод предназначен для решения двух основных задач (см. рис. 1.4): задачи обнаружения текста и задачи распознавания текста применительно к обнаруженным ТО. Следуя идее каскада фильтрации, которая состоит из последовательной обработки данных со все более и более селективными фильтрами, задача обнаружения текста выполняется в два этапа.

Первый этап – локализация ТО, целью которого является быстрое извлечение потенциальных текстовых блоков на изображениях (ТО-кандидатов) с очень низким коэффициентом потерь и разумным уровнем ложных тревог, и второй этап – верификация ТО-кандидатов на основе машинного обучения.

Показано, что для получения быстрого алгоритма, ТО-кандидаты локализуются на основании исследования эвристических характеристик. Автор алгоритма использует порог, чтобы отрегулировать недостатки классификатора на базе эвристических признаков при отделении ТО от фона. Низкое значение порога позволяет избежать пропуска текстовых блоков при локализации (низкий процент брака). Полученные при этом ложные тревоги (признание в

качестве ТО областей, которые таковыми не являются) удаляются на следующем шаге верификации. Тем не менее, слишком низкое значение порога дает много ложных ТО, при этом усложняется задача извлечения строки текста. Приемлемое значение порога может быть найдено на практике. На шаге верификации сначала выполняется нормализация размера строки-кандидата в текст. Затем используется классификатор на базе метода машинного обучения, многослойного персептрона и машин опорных векторов для отделения ТО от областей фона. Из-за большой дисперсии полутоновых значений текстовых символов, обучение классификаторов и верификация текстовых строк выполняется в пространстве признаков. Автор метода предлагает четыре таких признака, которые устойчивы к изменчивости значений интенсивностей символов на плутоновых изображениях.

Использование метода [43] обеспечивает высокую производительность локализации ТО в сочетании с более низкой вычислительной стоимостью по сравнению с другими методами за счет того, что процесс локализации позволяет избежать применения классификатора для всего изображения, а также снижает изменение размера текста путем выделения отдельных текстовых строк.

Несмотря на ряд достоинств метода [43], ему присущ и существенный недостаток, связанный с выделением эмпирических признаков и необходимостью подбора порога для классификации ТО-кандидатов, что снижает универсальность метода.

По характеру выделяемой информации методы обнаружения и локализации ТО на изображениях можно также объединить в три группы [54].

1. На основе использования контурной информации (каждый символ обладает четко выраженной контурной структурой); для локализации текста эта группа методов использует такие приемы как нахождение скелетона (skeletonization) [55], выделение краев (edge detection) и выделение углов (corner detection) [56], методы на основе инвариантных моментов (invariant methods)

[57] и пр. В случае изображений со сложным фоном быстрая обработка данных, полученных на этапе предобработки, представляет собой нетривиальную задачу.

2. На основе анализа цветовой информации (целевые текстовые области обладают однородными цветами/интенсивностью и удовлетворяют ограничениям по размеру и форме); чаще всего используются метод гистограмм (histogram method) [58], анализ связных компонент (connected component analysis CCA) [59] и различные алгоритмы адаптивной бинаризации (adaptive binarization) – в том числе алгоритмы Niblack, Sauvola, Chistian, Bernsan, Otsu и пр. [56]. Методы позволяют работать с произвольными размерами шрифта и произвольной направленностью текста, однако показывают плохие результаты на изображениях со сложным фоном, шумом и используют большое количество эвристических предположений.

3. На основе анализа текстурной информации (текстовые зоны могут заметно отличаться от фона, что позволяет использовать различные частотные фильтры для «пирамиды» изображений); для выявления нужных зон используется метод опорных векторов (support vector machines SVM), искусственные нейронные сети [60], экспертные системы и др. Методы позволяют работать с изображениями со сложным фоном, обладают высокой вычислительной сложностью из-за необходимости масштабирования изображений.

При разработке алгоритмов локализации ТО используется все многообразие методов обработки графической информации (рис. 1.5):

– в зависимости от размерности: одномерная (построчная, постолбцовая) либо двумерная обработка;

– по способу представления изображений: в пространственной либо спектральной области;

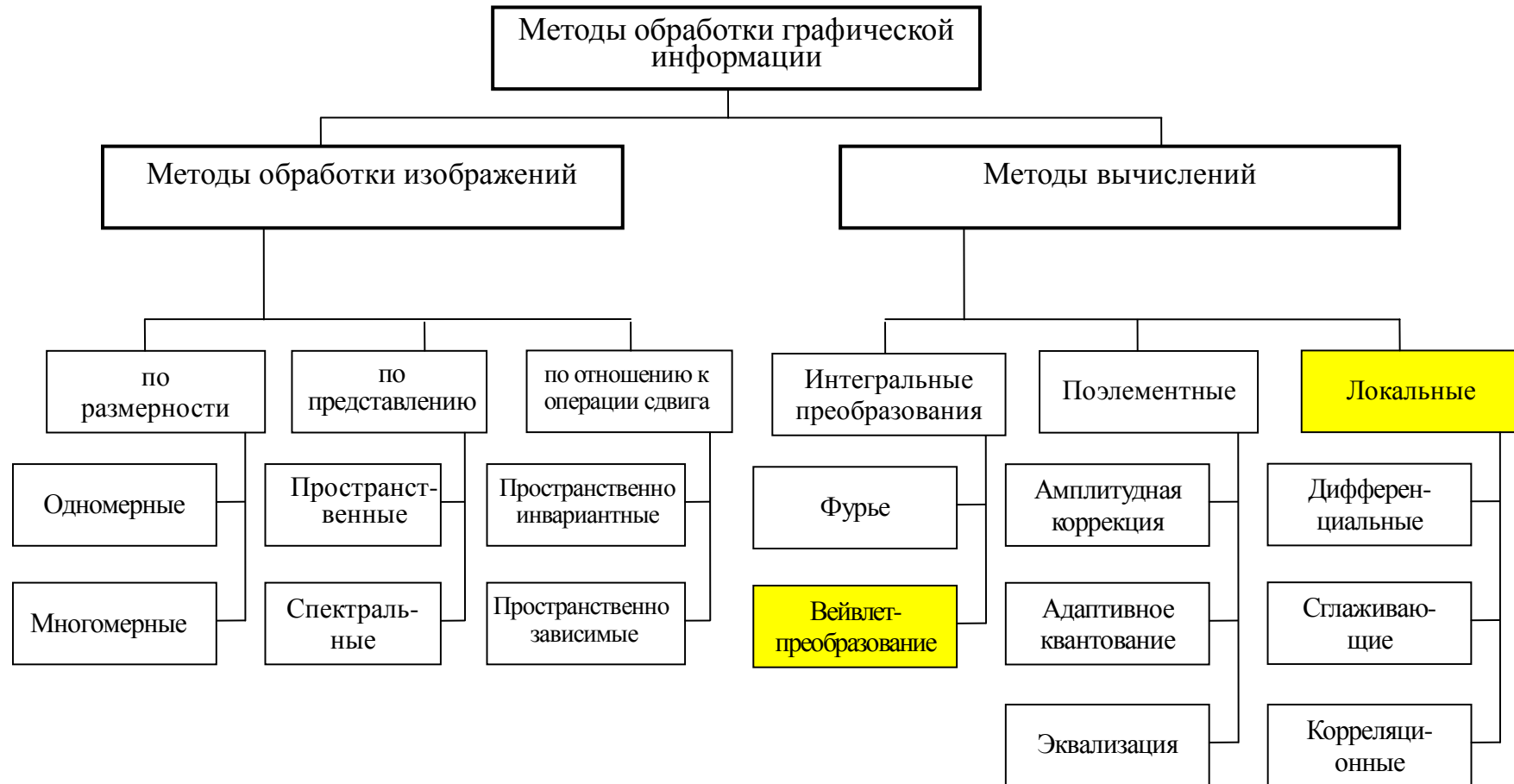


Рис.1.5. Классификация методов обработки графической информации.

- по отношению к операциям сдвига и масштаба: инвариантные или зависимые и др.

- в зависимости от методов вычислений возможны поэлементные операции (амплитудная коррекция, адаптивное квантование, эквализация), локальные операции над областями (дифференциальные, сглаживающие и корреляционные), глобальные и локальные интегральные операции (преобразования Фурье, непрерывное и дискретное вейвлет-преобразования и др.).

Как показывает анализ литературных источников, для локализации текстовых областей преимущества имеет локальная обработка и интегральные преобразования – вейвлет-преобразования.

Кроме того, интерес представляет классификация, учитывающая принципы выделения либо обнаружения текстовых областей, которые опираются на инструментарий распознавания образов, рассматривая обнаружение ТО как пример задачи бинарной классификации, когда входные образы нужно отнести к классу «текстовая область» или к классу «не текстовая область».

В частности, для обнаружения текстовых областей предложено [61] более детальное разделение методов на:

- основанные на знаниях (использование основных знаний человека о том, из чего состоит типичная ТО);
- основанные на инвариантных признаках (поиск структурных признаков ТО, которые существуют даже в случае изменения ракурса, условий освещения);
- основанные на сравнении с шаблоном (использование нескольких стандартных шаблонов для описания ТО в целом или ее отдельных признаков).

Рассмотрим более детально методы обнаружения текстовых областей.

1.3.1. Методы обнаружения текстовых областей, основанные на знаниях

Данные методы работают по нисходящему принципу и используют знания человека о типичном виде ТО, например:

- ТО имеет, как правило, повышенное значение дисперсии интенсивности пикселей по сравнению с фоном (текстурное описание)[62];
- в ТО наблюдается квазипериодические колебания интенсивности вдоль строки текста (пространственное контурное описание либо описание в спектральной области) [3, 4, 6, 46, 47, 48];
- существует диапазон соотношения высоты и ширины или другая информация о геометрических размерах ТО для конкретной задачи (например, при обнаружении номера автомобиля) и т.д. (атрибутивные параметры) [63, 64, 51].

Эти знания формализованы в виде некоторого набора правил (в соответствии со схемой формирования параметров изображения с ТО (рис. 1.6)) и последовательно применяются ко всем участкам изображения для поиска ТО.



Рис. 1.6. Схема формирования параметров изображения с ТО.

В частности, метод, представленный в [51, 64], для поиска ТО-кандидатов использует горизонтальные и вертикальные проекции входного изображения. Найденные кандидаты в дальнейшем верифицируются с помощью дополнительных классификаторов.

В общем, методы обнаружения ТО, основанные на знаниях, просты в реализации, основанные на понятных человеку правилах, и хорошо работают при локализации ТО на изображениях с несложным фоном. К недостаткам следует отнести сложность точного преобразования (формализации) человеческих знаний о ТО в правила, сложность обнаружения на одном изображении нескольких ТО с различными шрифтами и расположенных не параллельно по отношению к границам изображения, существенное снижение качества обнаружения ТО при наличии сложного фона на изображении, высокую вероятность ложного положительного обнаружения.

1.3.2 Методы обнаружения текстовых областей, основанные на инвариантных признаках.

Методы обнаружения ТО, основанные на использовании инвариантных признаков, работают по восходящему принципу, и базируются на предположении, что если человек может обнаружить ТО на изображении независимо от ракурса, индивидуальных особенностей, условий освещения, тогда должны существовать признаки присутствия ТО на изображении, инвариантные относительно условий наблюдения. К таким инвариантным признакам относят границы (контуры), значения яркости, форму, текстуру, цвет и другие [21, 62, 65, 66]. Эти признаки служат для поиска структурных элементов ТО. Верификация найденных ТО-кандидатов осуществляется с использованием дополнительных классификаторов на основе нейронных сетей, нечеткой логики и других [22, 67, 68, 69, 70].

В соответствии с требованиями к методам обнаружения ТО по инвариантным признакам для формального описания ТО необходимо

выделять как границы объектов (определять пространственные параметры объекта), так и оценивать характеристики областей (определять атрибутивные параметры) (см. рис. 1.6).

К пространственным параметрам относятся координаты объекта (или его частей), начальные геометрические характеристики (контуры, границы однородных сегментов, пространственные примитивы и производные элементы) изображения объекта распознавания. К оценке пространственных параметров предъявляются требования по точности (абсолютные и относительные ошибки) и помехоустойчивости.

К атрибутивным параметрам относятся цветовые, текстурные и гистограммные описания изображения в целом и составляющих его однородных сегментов. К оценке этих характеристик предъявляется требование полноты, т.е. соответствия полученных характеристик требованиям задачи. Атрибутивные характеристики дополняют геометрические данные [46].

Метод обнаружения объектов на изображении, предложенный в [64], использует карту контуров, полученную с помощью детектора Кэнни [71]. На основе правил контура группируются, и остаются только те кандидаты, которые соответствуют ТО.

В [72] разработан метод обнаружения объектов, который осуществляет поиск текстур, характерных для объекта. В оригинале метод рассчитан на поиск лиц на изображении, но его идея подходит и для обнаружения ТО. Параметры текстур вычисляются на основе статистических признаков второго порядка на каждом фрагменте изображения размером 16x16 пикселей. Также используется каскадная корреляционная нейронная сеть для классификации текстур и сеть Кохонена для формирования кластеров для разных текстурных классов.

Привлекательными с точки зрения простоты реализации и быстродействия являются методы с явным указанием границ кластера ТО. Процесс обнаружения ТО при этом состоит из следующих шагов:

- классификация пикселей входного изображения и получение нового изображения с отброшенными пикселями фона;
- сегментация с наращиванием регионов для областей пикселей с ТО;
- применение эвристических правил для отбрасывания областей фона областей и обнаружения ТО.

1.3.3. Методы обнаружения текстовых областей, основанные на сравнении с шаблоном

Использование методов обнаружения ТО, основанных на сравнении с шаблоном, требует наличия нескольких стандартных шаблонов для описания ТО в целом или ее отдельных признаков.

Основные требования к используемым шаблонам формулируются с учетом обеспечения с их помощью следующих возможностей методов обнаружения ТО на изображении:

- определение текстовых областей с учетом их фоновых заполнений;
- выделение текстовых областей с учетом разнообразия форм и размеров символов;
- определение границ текстовых областей с требуемой точностью.

После получения шаблона для обнаружения ТО применяются методы, основанные на классификации, и корреляционно-экстремальные методы. Их можно рассматривать как разновидности методов сравнения с шаблоном.

При использовании корреляционно-экстремального метода [44, 45] для обнаружения ТО для каждой области изображения и набора моделей ТО вычисляются взаимные корреляционные функции, определяется максимальное значение и принимается классификационное решение – «текстовая область» или «не текстовая область». Преимущества корреляционно-экстремального метода – хорошая помехоустойчивость в условиях сложного и неоднородного фона, учет пространственного расположения символов в ТО. А недостатки – большие вычислительные

затраты, большое количество моделей, большие затраты времени как в процессе разработки моделей, так и в процессе локализации, отсутствие инвариантности к масштабу изображения символов.

На рис. 1.7 представлена схема корреляционно-экстремального метода обнаружения ТО на изображении.

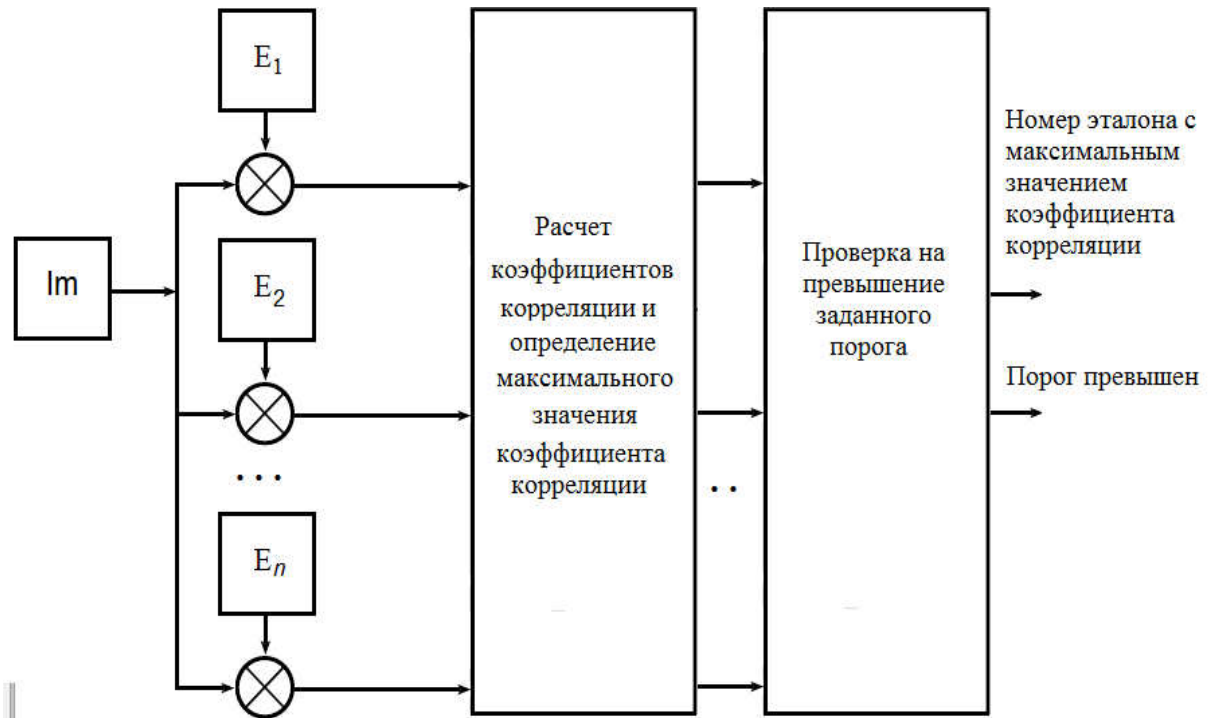


Рис. 1.7. Схема корреляционно-экстремального метода обнаружения ТО на изображении

При корреляционно-экстремальном методе рассматривается фрагмент изображения Im , для которого определяются коэффициенты корреляции с различными эталонными изображениями (E_1, E_2, \dots, E_n). Из полученных значений коэффициентов корреляции определяется максимальный и сравнивается с некоторым заранее заданным порогом. Значение этого порога определяет минимальную допустимую степень схожести изображений, ниже которой изображения априори не могут быть одинаковыми.

Проведенный анализ показывает, что в условиях сложного и неоднородного фона следует применять корреляционно-экстремальный

метод, как метод, основанный на сравнении с шаблоном. Однако такие его недостатки как большое количество моделей, большие затраты времени как в процессе разработки моделей, так и в процессе локализации ТО, отсутствие инвариантности к масштабу изображения символов снижают оперативность информационных систем переработки графической информации, использующих этот метод.

1.4. Особенности применения непрерывного вейвлет-преобразования при локализации текстовых областей.

Как показал анализ литературных источников и проведенные эксперименты, ТО на изображении имеют частотно-пространственные свойства, отличающиеся от аналогичных свойств сложного фона изображения. Поэтому для локализации ТО находят применение методы, основанные на частотно-избирательных свойствах объекта обнаружения, в частности, на основе вейвлет-преобразования.

Особенностью вейвлет-преобразования является возможность учитывать локальные пространственно-частотные свойства объектов и регулировать степень подчеркивания перепадов интенсивности на границах объектов разных размеров.

Для подчеркивания перепадов интенсивности отдельных символов с регулируемой степенью детализации применяют непрерывное вейвлет-преобразование (НВП) [46, 47, 48 73-79] строки изображения.

НВП функции $f(x) \in L^2(R)$ определяется путем свертки

$$W(s, x_0) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} f(x) \psi^* \left(\frac{x - x_0}{s} \right) dx ,$$

где $f(x) \in L^2$ – преобразуемая (анализируемая) функция;

$\psi^* \left(\frac{x-x_0}{s} \right)$ – двухпараметрическая базисная функция, получающаяся из материнского вейвлета $\psi_0(x)$ в результате масштабирования с масштабным уровнем $s \in R^+$ и сдвига с параметром $x_0 \in R$;

ψ^* – комплексно-сопряженная функция по отношению к ψ ;

s – масштабный уровень вейвлета;

x_0 – определяет положение вейвлета на оси x .

Множитель $\frac{1}{\sqrt{s}}$ вводится для того, чтобы все вейвлет-функции ψ_{s,x_0} имели постоянную норму в пространстве $L^2(R)$.

Для решения задачи подчеркивания перепадов интенсивности применяют действительные вейвелеты, заданные в виде симметричных нечетных функций $\psi(x) = -\psi(x)$, имеющие компактный или эффективный носитель, в частности, вейвелеты Гаусса, гиперболическое вейвлет-преобразование (ГВП) и т.д. [80-85]. Базисные вейвлет-функции должны удовлетворять ряду условий:

– условию локализации;

– условию допустимости: для Фурье-образа материнского вейвлета

$$C_\psi = \int_{-\infty}^{+\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty;$$

– условию осцилляции (знакопеременности)

$$\int_{-\infty}^{+\infty} \psi(x) dx = 0;$$

– условию ограниченности: норма базисной функции

$$\|\psi(x)\| = \sqrt{\int_{-\infty}^{+\infty} |\psi(x)|^2 dx} < \infty .$$

Несимметричные вейвлет-функции не используются для подчеркивания перепадов интенсивности изображения, т.к. смещают центр перепада интенсивности. В тоже время, симметричные вейвлет-функции обладают свойством сохранять неизменным положение центра перепада при подчеркивании перепадов интенсивности. Благодаря этому качеству они находят применение при локализации объектов, основанной на методах обнаружения резких изменений сигнала, таких как перепады интенсивности на изображении – методах выделения контуров [47, 80, 85, 86]. Это свойство симметричных вейвлет-функций полезно при обнаружении и локализации ТО на изображениях.

1.5. Особенности применения дискретного вейвлет-преобразования при локализации текстовых областей.

При обработке изображений и поиске ТО для практической реализации применяют дискретное вейвлет-преобразование (ДВП). Используется как одномерное ДВП (для отдельно взятой строки, интенсивность пикселей которой рассматривается как отсчеты некоторой решетчатой функции), так и двумерное ДВП для изображения в целом или некоторой его области [74, 76, 87]. Уровень разложения сигнала или области при ДВП на практике, как правило, не превышает 2.

Основная идея ДВП – такая же, что и при использовании НВП. Масштабно-временное представление сигнала получают с применением методов цифровой фильтрации. В случае выполнения НВП вычисляется корреляция между вейвлетом на разных масштабах и исходным сигналом. В дискретном случае для анализа сигнала на разных масштабах используются фильтры с различными частотами среза. Сигнал пропускается через древовидно соединенные высокочастотные (ВЧ) и низкочастотные (НЧ) фильтры. Для двумерного случая схема декомпозиции исходного изображения представлена на рис. 1.8.

В результате декомпозиции входного изображения на j -м уровне получают аппроксимирующие коэффициенты cA_j (сжатое в два раза изображение относительно входного) и детализирующие коэффициенты по направлениям (горизонтальные cD_j^g , вертикальные cD_j^v , диагональные cD_j^d).

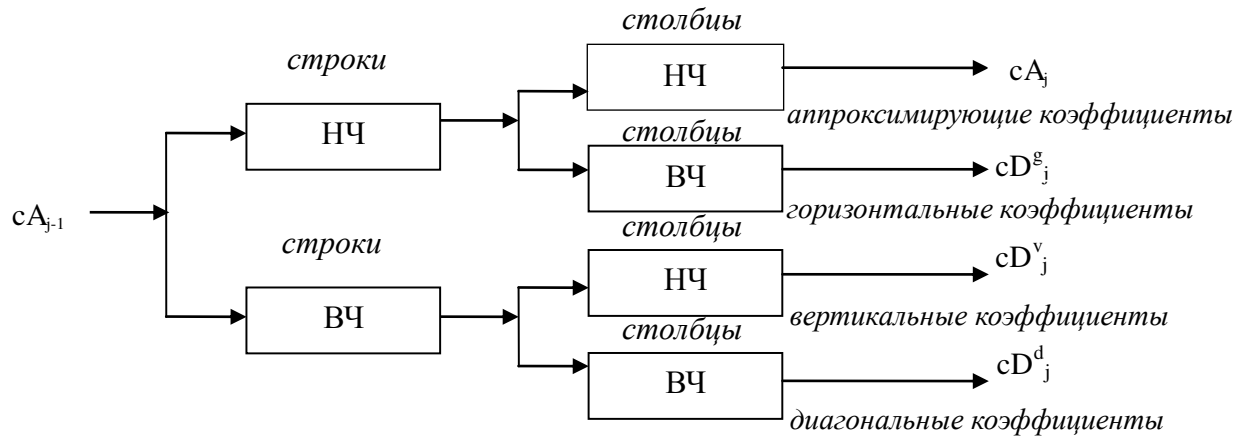


Рис. 1.8. Декомпозиция двумерного сигнала на один уровень.

В качестве входного выступает изображение в виде аппроксимирующих коэффициентов предыдущего уровня. На начальном этапе процесс ДВП начинается с использованием исходного изображения.

Как следует из схемы получения ДВП, входное изображение разлагается на четыре компоненты, соответствующие различным комбинациям последовательного использования НЧ и ВЧ фильтров, т.е. фактически получены четыре результата фильтрации за один уровень разложения. Полученные коэффициенты применяются для обнаружения ТО-кандидатов, например для нахождения контуров текстовых областей ТО. В качестве вейвлетов используют различные типы функций [74, 76].

Следует отметить, что традиционные фильтры обнаружения контуров могут обеспечить аналогичный хороший результат, но они не могут обнаружить три типа контуров (на каждом из изображений детализирующих коэффициентов) за один раз и требуют больше времени для обработки, чем двумерное ДВП.

1.6. Особенности использования свёрточных нейронных сетей для локализации текстовых областей

Для эффективной работы систем переработки графической информации метод локализации ТО на изображении, положенный в их основу, должен учитывать особенности представления данных, а именно – строгую двумерную структуру изображения ТО, в которой существует пространственная зависимость между значениями соседних пикселей.

Нейронные сети являются мощной и популярной технологией решения задач распознавания образов различной сложности. Наиболее часто в задачах распознавания и идентификации изображений используются классические нейросетевые архитектуры [88] (многослойный персептрон, сети с радиально базисной функцией и др.), но их применение для решения задачи локализации ТО на изображении неэффективно по следующим причинам [89]:

- изображения имеют большую размерность, соответственно возрастает размер нейронной сети [90];
- большое количество параметров увеличивает вместимость системы и соответственно требует большей тренировочной выборки, увеличивает время и вычислительную сложность процесса обучения;
- для повышения эффективности работы системы желательно применять каскады нейронных сетей (обученные с различными начальными значениями синаптических коэффициентов и порядком предъявления образов), но это увеличивает вычислительную сложность решения задачи и время выполнения [91, 92];
- отсутствует инвариантность к изменениям масштаба изображения, ракурсов съёмки камеры и других геометрических искажений входного сигнала [89].

Ян Лекун предложил использовать так называемые свёрточные нейронные сети (СНС), как аналог зрительной коры головного мозга для

распознавания изображений [93]. СНС – один из подвидов нейронных сетей, который учитывает особенности представления данных на изображениях [94 - 97]. Обучение СНС осуществляется с учителем

В основе построения СНС лежат три основные идеи:

– каждый нейрон получает входной сигнал от локального рецептивного поля в предыдущем слое, что обеспечивает локальную двумерную связность нейронов;

– каждый скрытый слой сети состоит из множества плоскостей (карт признаков), на которых все нейроны имеют общие веса, что обеспечивает инвариантность к смещению и сокращение общего числа весовых коэффициентов сети; в процессе обучения карты признаков выделяют различные признаки объекта.

– за каждым слоем свёртки следует вычислительный слой, осуществляющий локальное усреднение и подвыборку, что обеспечивает уменьшение разрешения для карт признаков.

Таким образом, набор плоскостей представляет собой карты характеристик, и каждая плоскость находит «свои» участки изображения в любом месте предыдущего слоя. Размер локального рецептивного поля выбирается самостоятельно в процессе разработки нейронной сети [93].

В свёрточных слоях при сканировании рецептивные поля частично наслаиваются друг на друга по принципу черепицы, в подвыборочных слоях области соседних нейронов не перекрываются. Подвыборочный слой уменьшает масштаб плоскостей путём локального усреднения значений выходов нейронов, таким образом, достигается иерархическая организация. Последующие слои извлекают более общие характеристики, меньше зависящие от искажений изображения. После прохождения нескольких слоев карта признаков вырождается в вектор [98].

СНС хорошо зарекомендовала себя для решения проблемы распознавания человека по фотопортрету или распознавания рукописных символов по их изображениям, но задача выбора оптимальных параметров

сети является открытой и плохо описанной в литературе.

Благодаря своей архитектуре СНС обладает устойчивостью к шумам, сдвигам, поворотам, масштабу входных образов, и продемонстрировала самую высокую достоверность в задачах распознавания лиц [99] и рукописных символов [93].

Разработчик СНС Я. Лекун предложил следующую архитектуру СНС (рис. 1.9) [93].

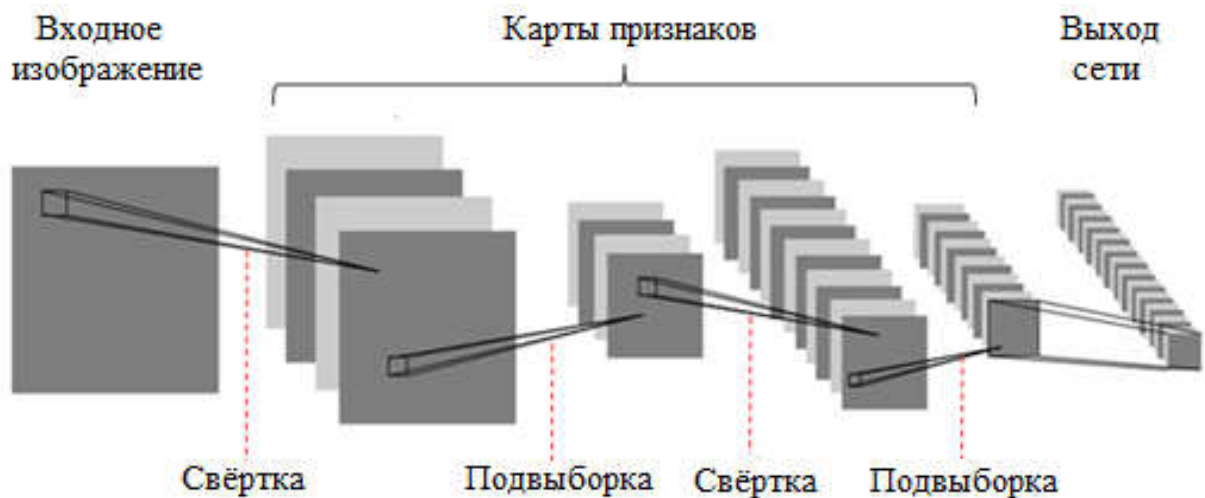


Рис. 1.9. Архитектура свёрточной нейронной сети Я. Лекуна.

К. Гарсиа та М. Делакис предложили метод обнаружения объектов [100, 101] (в оригинале человеческих лиц) и ТО [101] на основе СНС [93].

Обнаружение объектов на основе СНС происходит в две стадии: "грубое" (coarse) и "точное" (fine) обнаружения. Предварительная обработка входного изображения не происходит, но при этом в обучающую выборку СНС включаются примеры с разной степенью контрастности.

Особенностью метода является очень быстрая обработка изображения на первой стадии, когда входное изображение обрабатывается сетью за один этап, а не с помощью сканирования фиксированным окном, как в других методах на основе моделирования изображения лица или ТО. В результате "грубого" выявления определяются координаты объектов-кандидатов. На

следующем этапе происходит верификация кандидатов, когда они группируются и область изображения каждого из них проверяется с помощью СНС еще раз, но уже с использованием фиксированного окна, в соседних позициях и при нескольких уровнях масштабирования. При этом кандидат принимается при достижении порогового значения многократных обнаружений [101].

Данный метод верификации требует значительных временных затрат [102], так как для каждого кандидата необходимо осуществить около 27 симуляций СНС, если, например, проверяется только по одному соседнему окну при двух соседних уровнях масштабирования.

Рассмотренный метод в сочетании с предложенной архитектурой СНС, применяемый для распознавания рукописных символов [93] может рассматриваться как наиболее перспективный аналог при разработке методов обнаружения ГО на изображении.

На сегодняшний день разработано достаточно большое количество различных архитектур СНС для решения задач локализации и распознавания объектов на изображениях. Их ключевыми особенностями являются:

1) фиксированный размер входных данных (обычно используются размеры 16×16 пикселей, 28×28 пикселей, 29×29 пикселей, 32×32 пикселя, 36×64 пикселя);

2) методика передачи данных от слоя к слою (используется неполносвязный слой и предложена схема связей между картами слоев);

3) метод уменьшения размеров карты нейронов (использован шаг свертки, равный 2, за счет чего результирующая карта уменьшается в два раза; уменьшение размеров достигается за счет введения слоев подвыборки);

4) тип выходного слоя (выходной слой обычно представляет собой простой полносвязный слой на основе персептронов).

Разнообразные решения при построении СНС можно рассматривать, как попытки человека повлиять на признаки, которые извлекает сеть. В настоящий момент отсутствуют четкие рекомендации о выборе структуры

сети, количества карт в слое, количества карт предыдущего слоя, участвующих в формировании карты текущего слоя, количестве нейронов в полносвязном слое и т.д. Решения по структуре СНС, представленные в общедоступных источниках, являются вариациями архитектуры Я. Лекуна [93, 103, 104] с эмпирически подобранными параметрами для решения конкретных задач. Изменение условий задачи требует переобучения сети.

1.6.1. Особенности обучения свёрточной нейронной сети

Обучение нейронных сетей является одной из важных проблем при использовании их для решения различных задач переработки ГИ. Наличие качественного алгоритма обучения позволяет улучшить получаемые результаты, снизить ошибку построенной сети.

Для обучения СНС был выбран стандартный алгоритм обратного распространения ошибки (back propagation), который был предложен в 1986 году Румельхартом, Макклеландом и Вильямсом [105]. На выходе сети формируются значения +1 и -1 для изображения текстовой области и не текстовой области.

Значения исходных синаптических весов для всех слоёв сети с гиперболическим тангенсом выбираются на основе равномерного распределения с нулевым математическим ожиданием и дисперсией, обратной квадратному корню из количества синаптических связей нейрона [106, 107]. Алгоритм обратного распространения ошибки использует методику, позволяющую быстро вычислять вектор частных производных (градиент) сложной функции многих переменных, если структура этой функции известна. В качестве такой функции в алгоритме рассматривается функция ошибки сети и учитывается тот факт, что структура функции ошибки сети полностью определяется архитектурой нейронной сети, которая считается известной.

В самом начале обучения сеть не настроена, поэтому после

предъявления очередного входного образа, она формирует неточное выходное значение. Сравнение с желаемой реакцией дает возможность корректировать веса связей таким образом, чтобы сеть на следующем шаге могла выдавать более точный результат. Обучающее правило обеспечивает настройку весов связей. Информация о выходах сети является исходной для нейронов предыдущих слоев. Эти нейроны могут настраивать веса своих связей для уменьшения погрешности на следующем шаге.

Функция ошибки представляет собой разность между текущим выходом сети и идеальным выходом, который необходимо получить. Для успешного обучения сети требуется приблизить выход сети к желаемому выходу, т. е. последовательно уменьшать величину функции ошибки. Это достигается настройкой межнейронных связей. Каждый нейрон в сети имеет свои веса, которые настраиваются, чтобы уменьшить величину функции ошибки. Выражение для коррекции синаптических коэффициентов имеет вид:

$$w_{ij}(t + 1) = w_{ij}(t) + \rho \frac{dE}{dw_{ij}},$$

где $w_{ij}(t)$ и $w_{ij}(t + 1)$ – вес связи между i -м и j -м нейронами на текущем и последующем шаге обучения,

$\frac{dE}{dw_{ij}}$ – производная функции ошибки,

ρ – параметр скорости обучения.

На каждой итерации алгоритма весовые коэффициенты нейронной сети модифицируются так, чтобы улучшить решение одного обучающего примера. При обучении свёрточной нейронной сети настройка весовых коэффициентов методом обратного распространения ошибки имеет особенности для разных слоев (полносвязных, подвыборочных, свёрточных). Подробно особенности обучения СНС, используемых в данной работе, рассмотрены в разделе 3.

Проведенный анализ показал, что одной из основных проблем, с которой сталкиваются при обучении СНС для обнаружения локализации ТО

на изображениях со сложным фоном, является то, что качество принятия классификационного решения о характере области изображения (текстовая область, не текстовая область) существенно зависит от объема и представительности обучающей выборки.

В настоящий момент для обучения СНС и других типов классификаторов для обнаружения и локализации ТО используются специализированные базы изображений (например, база данных конкурса алгоритмов искусственного интеллекта, ICDAR 2011 Robust Reading Competition [108]). Текстовые символы, содержащиеся в этих изображениях, имеют различные цвета, оттенки для полутоновых изображений, шрифты, размеры. Некоторые символы являются прозрачными и оформлены с контурами. Сами текстовые области расположены на различном, в том числе и сложном фоне (см. рис. 1.2, рис. 1.3).

Однако использование таких баз не устраняет проблему формирования обучающих выборок, которая связана со следующими факторами:

- большой объем обучающей выборки, позволяющий учесть максимальное количество реальных вариантов ТО, обеспечивает улучшение работы классификатора, но значительно увеличивает время обучения, что часто является недопустимым;
- сложность учета особенности формы представления текстовых фрагментов на изображениях (т.е. типы шрифта, размер шрифта, вид представления, угол расположения и др.);
- качество выбранных изображений, содержащих ТО;
- учет опыта специалиста, приводящий к субъективно сформированной обучающей выборке.

Поэтому в работе ставится задача разработки метода формирования ТО и использование его для автоматического формирования обучающей выборки, чтобы устранить влияние субъективного подхода человека при ее создании.

1.7. Методы оценки качества обнаружения и локализации текстовых областей на изображении

В процессе локализация ТО сначала происходит обнаружение ТО на изображении, а потом уже собственно локализация ТО (определение координат границ ТО). Поэтому, в конечном итоге погрешность локализации зависит от качества обеих процедур, для каждой из которых используются свои оценки качества выполнения процедуры.

Для оценки качества обнаружения текстовых областей использованы меры точности Precision (Pr) и полноты Recall (Rc) [109, 110].

Мера точности (Precision) определяется

$$Pr = (OTP / OP) \times 100\%, \quad (1.3)$$

где *OTP* – количество правильно обнаруженных текстовых областей;

OP – области, которые обнаружены как текстовые, хотя среди них есть и нетекстовые.

Мера полноты Recall определяется

$$Rc = (OTP / (OTP + НОТР)) \times 100\%, \quad (1.4)$$

где *OTP* – количество правильно обнаруженных текстовых областей;

НОТР – количество текстовых областей, которые не обнаружены.

Ошибка первого рода (вероятность ложноположительной локализации ТО) определяется как соотношение количества предъявлений объекта системе, когда тест ложно сообщает положительный результат, к общему числу изображений, не содержащих объекта.

Ошибка второго рода (вероятность ложноотрицательной локализации ТО) определяется как соотношение ложно отвергнутых предъявлений к общему числу предъявлений объекта системе

$$ERRI = \frac{NFTO}{FTO + NFTO} ,$$

$$ERRII = \frac{FNTO}{FTO + FNTO} ,$$

где $ERRI$ – ошибка первого рода (вероятность ложноположительной локализации ТО);

$ERRII$ – ошибка второго рода (вероятность ложноотрицательной локализации ТО).

FTO – количество найденных ТО (правильная локализация ТО),

$NFTO$ – количество найденных областей, которые не содержат текст (ложная локализация ТО),

$FNTO$ – количество ТО, которые система не нашла.

Для оценки качества локализации ТО используется метрический показатель близости между множеством координат эталонных границ I_{et} ТО тестового изображения и множеством координат границ I_t локализованной ТО, полученными в результате применения рассматриваемого метода локализации (критерий FOM (Figure of Merit), предложенный Прэттом) [85]

$$F_{loc} = \frac{1}{\max(N_t, N_{et})} \sum_{i=1}^{N_t} \frac{1}{1 + d^2(i)} , \quad (1.5)$$

где N_{et} – количество пикселей в множестве I_{et} ;

N_t – количество пикселей в множестве I_t ;

$d^2(i) = \min(I_t(i) - I_{et}(j))^2$, $j = \overline{1, N_t}$ – квадрат расстояния от i -го пикселя

множества I_t к ближайшему пикселю множества I_{et} .

1.8. Выводы. Постановка задачи исследований

Проведенный анализ информационных технологий переработки графической информации показал, что для широкого круга специализированных изображений (медицинские, спутниковые и другие), фотографии, сканированные документы, видео-файлы и т.д. важной задачей является распознавание текста на изображениях со сложным фоном (текст на рекламных щитах, номера домов, надписи на одежде, автомобилях и т.д.), эффективность решения которой зависит от качества локализации текстовых областей.

Показано, что процесс локализации текстовых областей предусматривает выполнение двух основных процедур: выявление текстовой области и ее верификацию. Проведенный анализ информационных технологий локализации текстовых областей позволил выделить три основных подхода к реализации процедуры локализации ТО: корреляционно-экстремальный, на основе сегментации спектральных текстур и структурный.

Показано, что несмотря на высокое качество локализации и учет пространственного расположения символов в ТО недостатками корреляционно-экстремального подхода являются неопределенность выбора модели в случае сложного фона, отсутствие инвариантности к масштабу изображения символов, значительные вычислительные и временные затраты, как в процессе разработки моделей, так и в процессе локализации ТО.

При использовании для локализации текстовых областей подхода на основе сегментации спектральных текстур не учитывается пространственное расположение символов, не обеспечивается достаточная помехоустойчивость в условиях сложного и неоднородного фона. При использовании структурного подхода несмотря на возможность выявления и некоторого учета пространственного расположения структурных примитивов к низкой помехоустойчивости в условиях сложного и неоднородного фона добавляется отсутствие инвариантности к масштабам изображения символов.

Таким образом, проведенный анализ показал, что при наличии сложного и неоднородного фона на изображениях лучшие результаты при оценке качества локализации ТО принадлежат методам на основе корреляционно-экстремального подхода. Однако, их недостатки – большое количество моделей, большие затраты времени как в процессе разработки моделей, так и в процессе локализации ТО, отсутствие инвариантности к масштабу изображений символов – снижают оперативность систем переработки графической информации, использующих эти методы. В результате возникает противоречие между качеством локализации ТО и оперативностью этой процедуры, особенно при наличии сложного фона на обрабатываемых изображениях.

В связи с этим, сформулирована цель диссертационного исследования – повышение качества и оперативности локализации текстовых областей на изображениях со сложным фоном в информационных технологиях переработки графической информации.

Для реализации поставленной цели предложено использовать корреляционно-экстремальный метод обнаружения ТО и для его реализации свёрточные нейронные сети, особенностью которых является наличие слоя представления фрагмента входного изображения в пространстве дискретного вейвлет-преобразования.

С учетом этого сформулированы дальнейшие задачи исследований.

Для достижения поставленной цели в работе необходимо решить следующие *основные задачи*:

- создать модель представления символа текста на изображении;
- разработать метод формирования текстовых областей на изображениях путем использования вероятностной модели представления символа текста при формировании обучающей выборки для обучения классификатора;
- разработать метод локализации текстовых областей на изображениях со сложным фоном на основе свёрточной нейронной сети.

– разработать на основе предложенных модели и методов информационную технологию локализации текстовых областей на изображениях для автоматической переработки графической информации.

РАЗДЕЛ 2

**МОДЕЛИ ПРЕДСТАВЛЕНИЯ И МЕТОДЫ ФОРМИРОВАНИЯ
ТЕКСТОВОЙ ОБЛАСТИ НА ИЗОБРАЖЕНИИ В СИСТЕМАХ
ПЕРЕРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ**

2.1. Обобщенная схема информационной технологии локализации текстовых областей на изображениях

С учетом рассмотренного материала в разделе 1 (рис. 1.4) разработана обобщенная схема ИТ локализации ТО на изображениях на основе двухуровневого подхода (рис. 2.1).

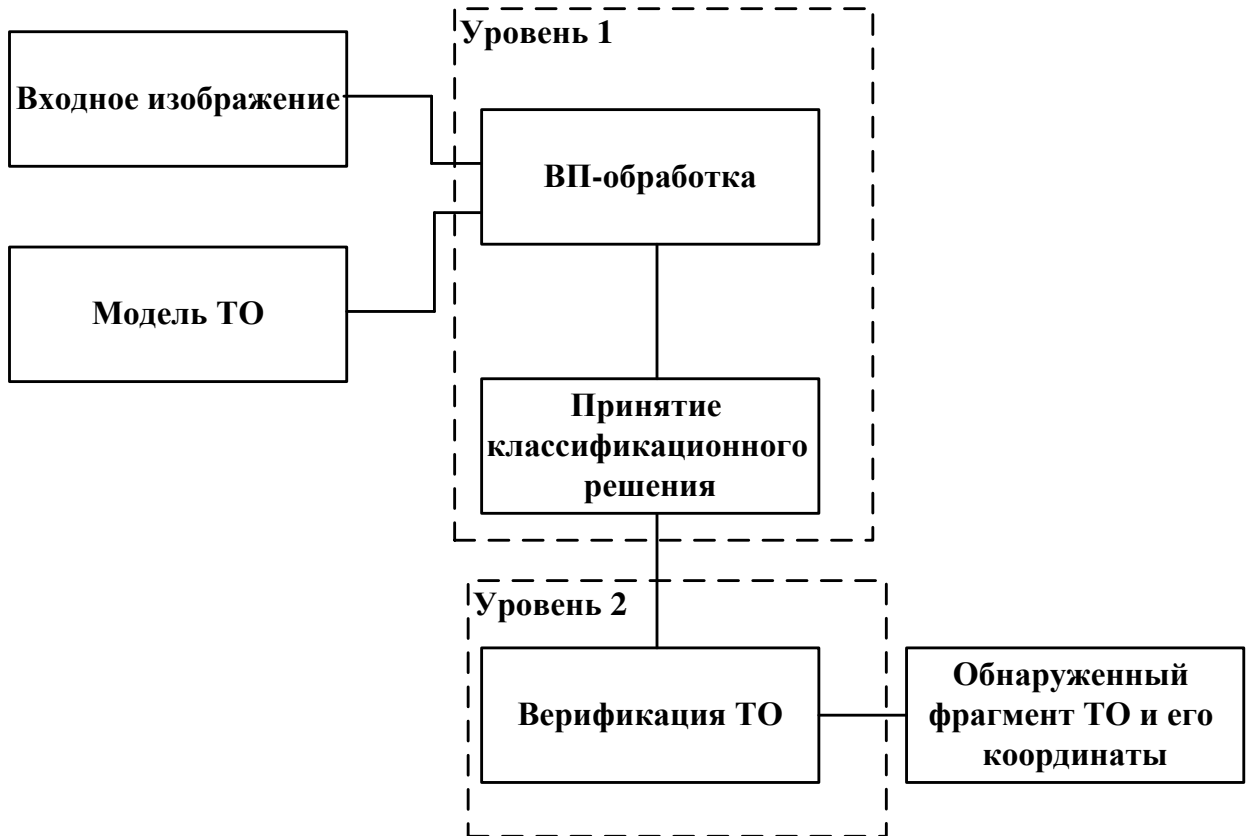


Рис. 2.1. Обобщенная схема ИТ локализации ТО.

В основе предлагаемой ИТ лежит усовершенствованный метод локализации ТО на изображении, который использует вейвлет-преобразование [14-15] и классификацию для отбора ТО-кандидатов.

Метод реализуется следующим алгоритмом:

Уровень 1 (обнаружение ТО).

Шаг 1. Формируется модель представления ТО и выполняется ее вейвлет-преобразование.

Шаг 2. Проводится вейвлет-преобразование и исходного изображения или его фрагмента.

Шаг 3. На основе результатов корреляционно-экстремального анализа вейвлет-преобразования модели ТО и исходного изображения или его фрагмента принимается классификационное решение.

Шаг 4. Повторяются шаги алгоритма 1 – 3 (для всего изображения).

Уровень 2 (верификация отобранных областей-кандидатов).

Шаг 5. Отобранные области подвергаются верификации по форме и размерам. Формируется список координат текстовых областей, содержащих символы.

Как было показано выше, качество принятия классификационного решения о характере области изображения (текстовая или не текстовая) существенно зависит от модели представления ТО.

При создании модели представления ТО учитывают следующее:

– для читабельности текста отдельные символы надписи должны иметь достаточно четкие очертания. Поскольку, текстовые строки имеют «сильные» вертикальные контуры, которые расположены в горизонтальном направлении с высокой плотностью, можно определить текстовую последовательность (в строке изображения) как квазипериодическую последовательность импульсов. Именно эта особенность надписи обычно используется при решении задачи обнаружения текстовых областей на изображении.

– нахождение фрагментов изображения с ТО сводится к выделению, во-первых, границ отдельных символов, во-вторых, границ ТО в целом и передачи координат определенной области для дальнейшего анализа (идентификации и классификации).

В рамках данной работы рассмотрены два подхода к построению

модели ТО:

– одномерный; ТО рассматривается как некоторый набор строк изображения, которое содержит символы (рис. 2.2, а).

$$u = f(x),$$

где x – положение пикселя в строке $x = \overline{1, L}$, L – длина строки изображения в пикселях.

Каждая из таких строк (рис. 2.2, а). может быть промоделирована квазипериодической последовательностью (рис. 2.2, б).

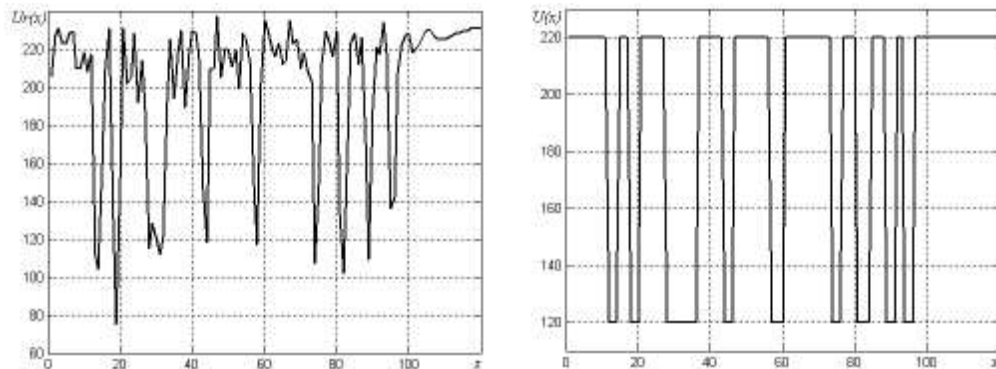


Рис. 2.2. Фрагмент строки с текстом:

а – для реального изображения; б – для модели в виде квазипериодической последовательности.

– двумерный; ТО рассматривается как некоторый набор областей изображений символов.

$$u = f(x, y),$$

где x, y – горизонтальные и вертикальные координаты области изображения символа;

$x = \overline{1, L}$, L – ширина области изображения символа, пикселей;

$y = \overline{1, M}$, M – высота области изображения символа, пикселей.

Каждая из таких областей может быть промоделирована с помощью вероятностной модели представления символов.

В данном разделе рассмотрены особенности формирования этих моделей.

Вопросы принятия классификационного решения будут обсуждены в разделе 3.

2.2. Квазипериодическая модель строки изображения, которое содержит символы текста

Для локализации надписи на изображении, как правило, применяют одномерную построчную (либо постолбцовую) обработку, что позволяет обрабатывать функцию интенсивности в строке (столбце) как одномерный сигнал, модель которого можно представить в виде [46]

$$f(x) = \sum_{i=1}^{1=N} \left(1 \left(x - x_0 - (i-1)T - \sum_{k=2}^i \varepsilon_{Tk} \right) - 1 \left(x - x_0 - (i-1)T - \sum_{k=2}^i \varepsilon_{Tk} - d - \xi_i \right) \right), \quad (2.1)$$

где N – количество прямоугольных импульсов последовательности;

$1(t)$ – единичная функция Хевисайда;

x_0 – координата первого импульса;

ε_{Tk} и ξ_i – случайные изменения периода T и длительности d импульса соответственно.

Пример моделирования строки текста согласно (2.1) приведен на рис. 2.3.

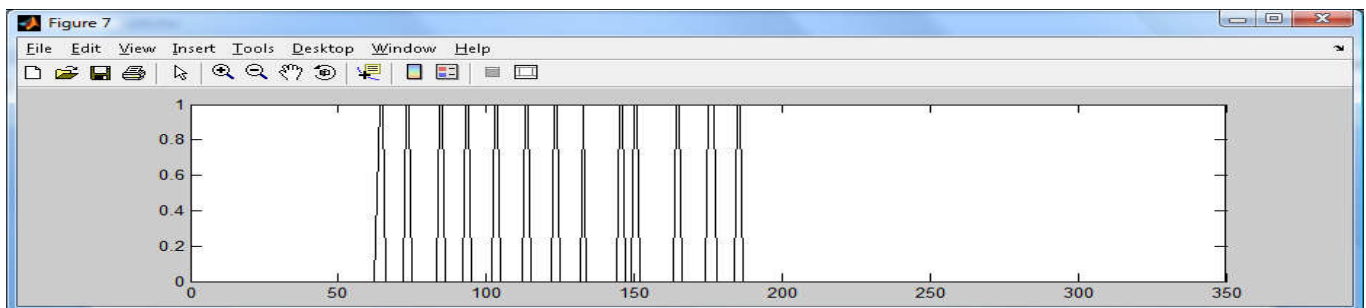


Рис. 2.3. Модель строки изображения с текстом:

Период и длительность наблюдаемых импульсов зависит от типа используемого шрифта. Для определения вида и параметров распределения периода T и длительности d импульсов были отсканированы и обработаны несколько страниц с текстом со шрифтами различных размеров (14,16,18,20) и типов (Arial, Times New Roman, Calibri, Cambria).

В качестве оценок использовались математические ожидания периода следования импульсов $M(T)$ и их длительности $M(d)$. Импульсы определялись при рассмотрении строки текста соответствующего шрифта как результат смены значения интенсивности с 0 на 1. Для уменьшения влияния абсолютного размера шрифта на качество дальнейшей локализации целесообразно определить коэффициент заполнения $K = M(d)/M(T)$.

Результаты статистической обработки представлены на рис. 2.4. и в табл. 2.1.

Для обобщения статистических данных и получения математической модели получены аппроксимирующие зависимости плотности распределения периода T , длительности d и коэффициента заполнения K для усредненных показателей шрифта.

Эмпирические значения соответствующих величин изображены в виде гистограммы, а непрерывные линии получены по аппроксимирующим выражениям

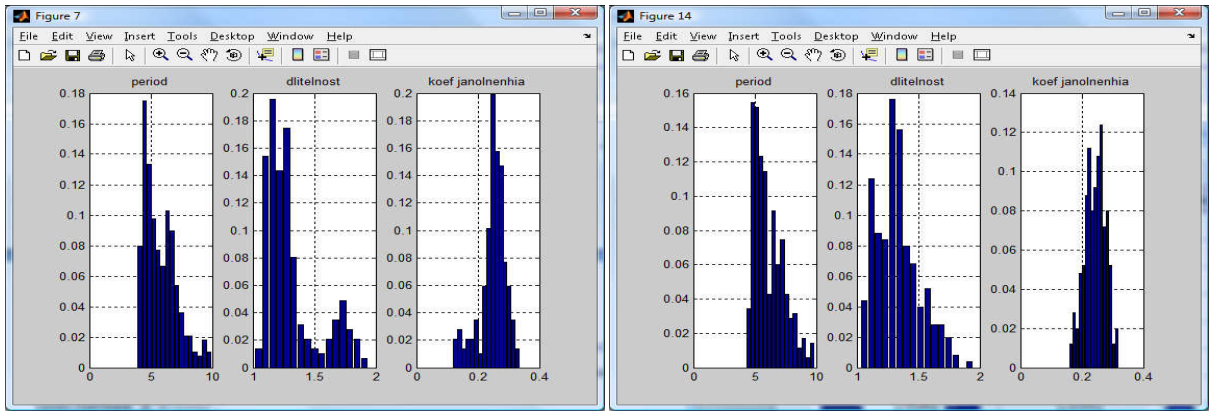
$$p_T(x) = \begin{cases} 1,25(e^{-0,5(x-3,77)} - e^{-(x-3,77)}), & x \geq 4; \\ 0, & x < 4; \end{cases}$$

$$p_d(x) = \begin{cases} 0,32e^{-0,4(x-1)}, & x \geq 1; \\ 0, & x < 1; \end{cases}$$

$$p_K(x) = 0,144e^{-200(x-0,233)^2}.$$

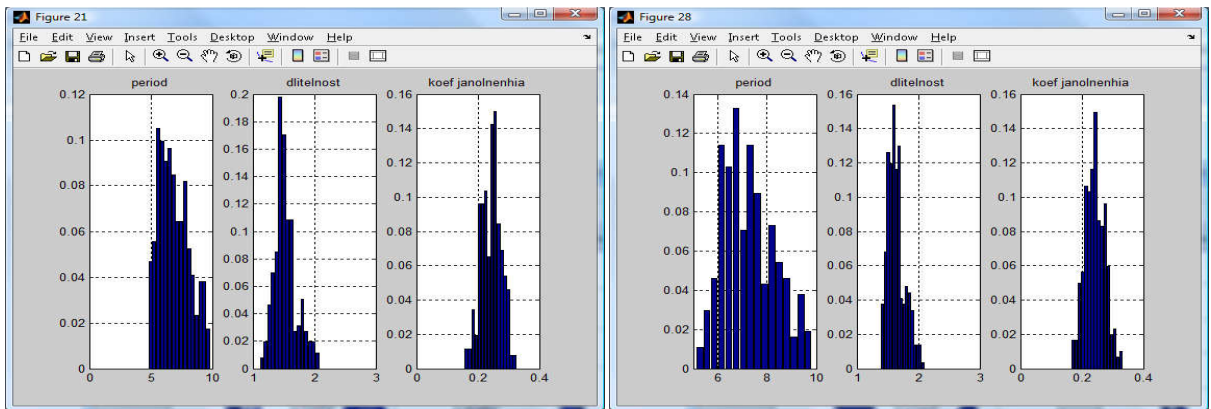
где $p_T(x)$, $p_d(x)$, $p_K(x)$ – оценка плотностей вероятностей периода, длительности и коэффициента заполнения импульсов соответственно.

Результаты представлены на рис. 2.5.



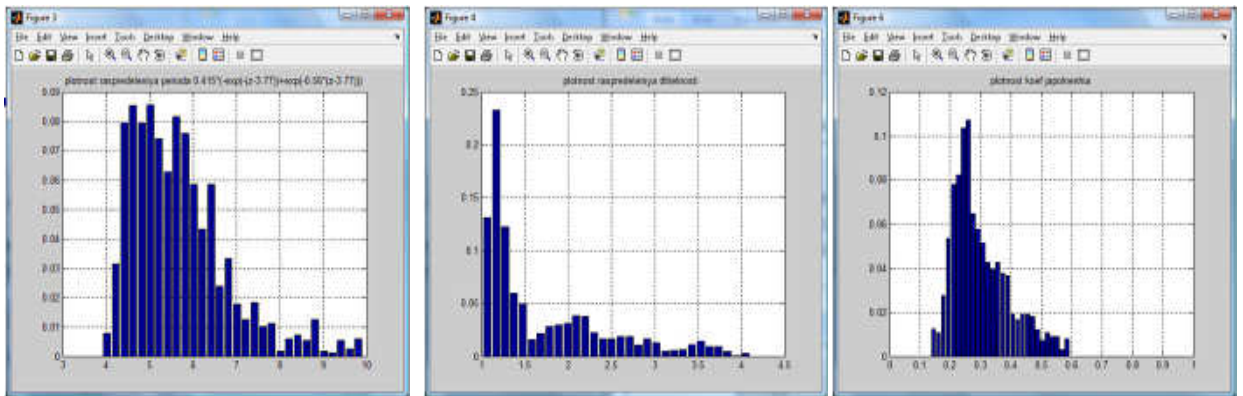
а)

б)



в)

г)



д)

е)

ж)

Рис. 2.4. Гистограммы полученных значений $M(T)$, $M(d)$ и $K = M(d)/M(T)$ с текстом Times New Roman: а – с шрифтами размера 14; б – с шрифтами размера 16; в – с шрифтами размера 18; г – с шрифтами размера 20; плотности распределения для текстов с шрифтами размера 14: д – периода T ; е – длительности d ; в – коэффициента заполнения K .

Таблица 2.1 – Статистические параметры для текстов с шрифтами разных размеров(14,16,18,20) типов (Arial, Times New Roman, Calibri, Cambria)

Тип шрифта	Параметры	Размеры шрифта			
		14	16	18	20
Times New Roman	$M(T)$	5,777	6,118	6,896	7,253
	$M(d)$	1,307	1,345	1,537	1,658
	$K = M(d)/M(T)$	0,226	0,220	0,223	0,229
Arial	$M(T)$	5,419	6,429	7,410	7,896
	$M(d)$	1,630	1,590	2,709	2,697
	$K = M(d)/M(T)$	0,301	0,247	0,365	0,342
Calibri	$M(T)$	5,590	5,947	6,798	7,168
	$M(d)$	1,171	1,918	1,980	2,037
	$K = M(d)/M(T)$	0,211	0,223	0,291	0,284
Cambria	$M(T)$	6,086	6,339	7,072	7,594
	$M(d)$	1,175	1,470	1,591	1,913
	$K = M(d)/M(T)$	0,193	0,232	0,225	0,252
Усредненный шрифт	$M(T)$	5,688	6,210	7,049	7,438
	$M(d)$	1,710	1,890	2,228	2,279
	$K = M(d)/M(T)$	0,301	0,304	0,313	0,306

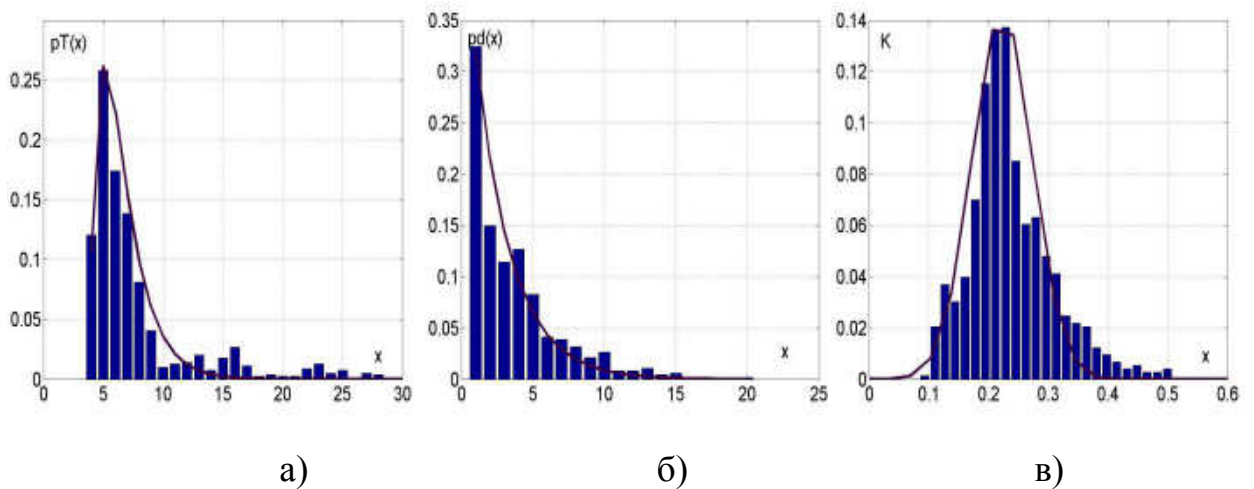


Рис. 2.5. Плотности распределения периода T (а), длительности d (б), и коэффициента заполнения K (в) импульсов в строке с текстом.

Анализ полученных зависимостей показал: 1) закон распределение периода импульсов существенно отличается от нормального; 2) закон распределения длительности имеет сходство с распределением Парето; 3) закон распределения коэффициента заполнения может быть нормальным с математическим ожиданием $M(K)=0,223$ и среднеквадратическим отклонением $\sigma = 0,05$. Математическое ожидание коэффициента заполнения в проведенных экспериментах находится в узком диапазоне значений $M(K)=0,20...0,32$, что позволяет использовать этот параметр в модели (2.1) как инвариантную величину к размеру шрифта.

Полученные статистические параметры использованы при обучении различных классификаторов и моделировании алгоритмов локализации символьных надписей согласно модели (2.1). Проведенные эксперименты подтвердили адекватность модели строки и реальных значений интенсивности в строке изображения. Однако, в рассмотренных моделях имеется существенный недостаток – не учитываются пространственные связи, присутствующие на изображении текстовых областей, что снижает качество обнаружения ТО.

2.3 Исследование квазипериодической последовательности в области вейвлет-преобразований

Рассмотрим поведение квазипериодической последовательности (см. формулу (2.1)) в области вейвлет-преобразования (ВП) [2, 10, 46, 47]: .

Как известно, вейвлет-преобразование функции $f(t)$ при действительных вейвлет-функциях (ВФ) определяется выражением [73-76]:

$$WF(x, s) = \int_{-\infty}^{+\infty} f(t) \psi\left(\frac{x - x_0}{s}\right) dx, \quad (2.2)$$

где $\psi\left(\frac{x-x_0}{s}\right)$ – базисная ВФ с единичной нормой для всех масштабов s ;

x – параметр сдвига.

Поскольку аналитические выкладки в пространственной области слишком громоздки, то анализ непрерывного ВП $WF(x, s)$ проведем в частотной области, используя Фурье-образ $W\hat{F}(\omega, s)$ [47, 48].

С учетом свойств преобразования Фурье выражение (2.2) имеет вид:

$$W\hat{F}(\omega, s) = \Phi(f(t)) \cdot \Phi(\psi(t, s)) = \hat{F}(\omega) \cdot \hat{\psi}(\omega, s),$$

где $\Phi(\cdot)$ – оператор преобразования Фурье,

$\hat{F}(\omega) \cdot \hat{\psi}(\omega, s)$ – Фурье-образы $f(t)$ и $\psi(t, s)$ соответственно.

Функция интегрального распределения энергии коэффициентов ВП по масштабам (ИРЭМ) в частотной области с учетом Фурье-образа $W\hat{F}(\omega, s)$ вейвлет-преобразования $WF(x, s)$ определяется в общем случае

$$E_N(s) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |W\hat{F}(\omega, s)|^2 d\omega.$$

Тогда с учетом четности подынтегрального выражения и характера анализируемой функции (2.1), описывающей последовательность прямоугольных импульсов, получим

$$E_N(s) = \frac{4}{\pi} \int_{-\infty}^{+\infty} |\hat{\psi}(\omega, s)|^2 \frac{\sin^2 \frac{\omega d}{2} \cdot \sin^2 \frac{N\omega T}{2}}{\omega^2 \sin^2 \frac{\omega T}{2}} d\omega.$$

Можно получить аналитические выражения для функции ИРЭМ при конкретных значениях N .

Для $N = 1$ (модель строки с одиночным объектом):

$$E_1(s) = \frac{4}{\pi} \int_0^{+\infty} |\hat{\psi}(\omega, s)|^2 \frac{\sin^2 \frac{\omega d}{2}}{\omega^2} d\omega = I_0.$$

Аналогично для N=2

$$E_2(s) = \frac{4}{\pi} \int_0^{+\infty} |\widehat{\Psi}(\omega, s)|^2 \frac{\sin^2 \frac{\omega d}{2} \cdot \sin^2(\omega T)}{\omega^2 \sin^2 \frac{\omega T}{2}} d\omega = 2I_0 + 2I_1.$$

где I_1

$$= \frac{4}{\pi} \int_0^{+\infty} |\widehat{\Psi}(\omega, s)|^2 \frac{\sin^2 \frac{\omega d}{2}}{\omega^2} \cos(\omega T) d\omega.$$

При обобщении полученных математических выражений при различных значениях длин последовательности N с учетом следующего обозначения

$$I_i = \frac{4}{\pi} \int_0^{+\infty} |\widehat{\Psi}(\omega, s)|^2 \frac{\sin^2 \frac{\omega d}{2}}{\omega^2} \cos(i\omega T) d\omega,$$

получены результаты, позволяющие представить ИРЭМ вейвлет-преобразования в виде $E_N(s) = \sum \alpha_i I_i$, где α_i – некоторые коэффициенты (рис. 2.6.).

$E_N(s) = \sum \alpha_i I_i$					
N	I_0	I_1	I_2	I_3	I_4
	α_0	α_1	α_2	α_3	α_4
1	1				
2	2	2			
3	3	4	2		
4	4	6	4	2	
5	5	8	6	4	2

Рис. 2.6. Коэффициенты функции ИРЭМ

Анализ полученных коэффициентов позволил получить математическое описание ИРЭМ ВП квазипериодической последовательности импульсов, имеющих длину N , в виде

$$E_N(s) = NI_0 + 2 \sum_{i=1}^{N-1} (N-i)I_i.$$

Полученное выражение справедливо для действительных симметричных нечетных вейвлет-функций.

Определены ИРЭМ для основных ВФ с учетом относительного масштаба $z = \frac{s}{T}$ и коэффициента заполнения $k = \frac{d}{T}$. Выполнен анализ на экстремум полученных аналитических выражений. Показано, что для ВП модели строки с использованием ВФ с негладкими передаточными характеристиками (ВП Хаара, ГВП (рис. 2.7)) функция ИРЭМ носит многоэкстремальный характер (рис. 2.8, а, б), а для ВФ с гладкими передаточными характеристиками (ВП Гаусса рис. 2.9) – одноэкстремальный характер (рис. 2.10, а, б).

В первом случае количество экстремумов зависит от масштаба наблюдаемых импульсов, коэффициента заполнения и количества импульсов N в последовательности. Для каждого N существует такое максимальное значение k , при превышении которого экстремальность не наблюдается, ИРЭМ приобретает монотонно возрастающий характер. Точка первого максимума ИРЭМ соответствует резонансному масштабу ВФ, при котором подчеркиваются границы объектов с наибольшей детальностью. Координаты последующих точек максимума соответствуют изменению степени подчеркивания границ объектов или изменению детальности. Наблюдаемая точка последнего минимума соответствуют масштабу ВФ, при котором последовательность воспринимается как единый объект, т.е. подчеркиваются границы всей последовательности.

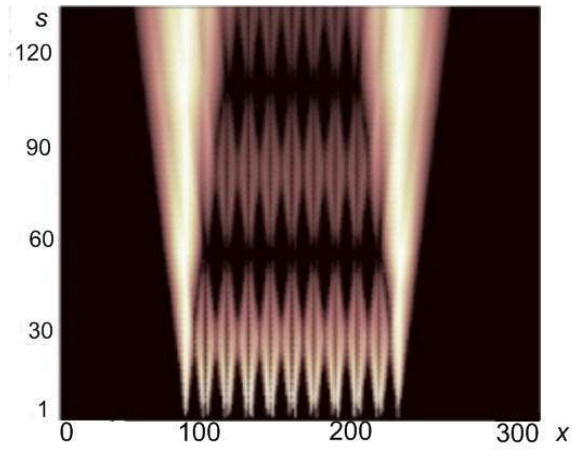


Рис. 2.7. ВП Хаара модели строки.

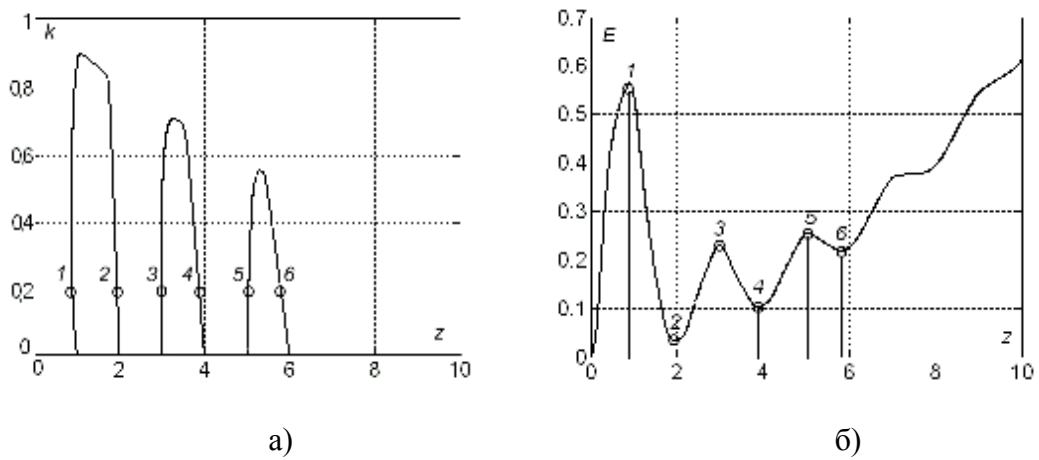


Рис. 2.8. Результаты расчета ИРЭМ для ВФ Хаара:

а – зависимость координат экстремумов ИРЭМ;

б – вид ИРЭМ при $k=0.2$.

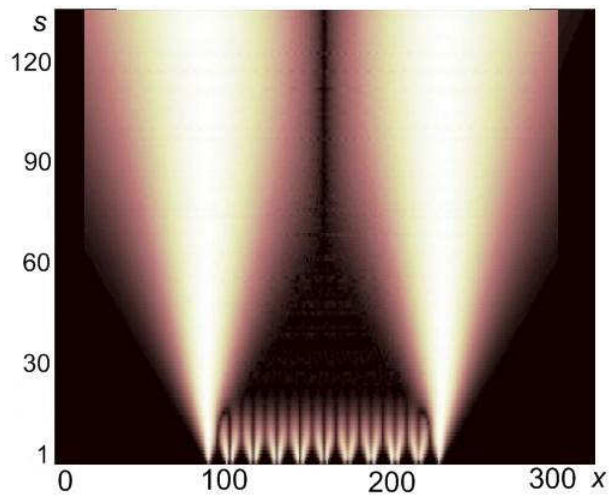


Рис. 2.9. ВП Гаусса модели строки.

Исследованы аналогичные зависимости ИРЭМ для ВФ с гладкими передаточными характеристиками (функция Гаусса) (см. рис. 2.9, рис. 2.10).

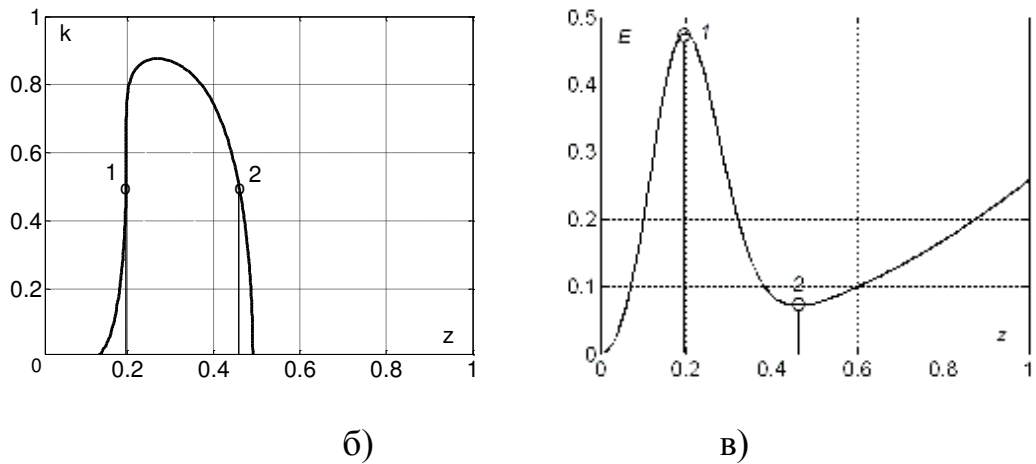


Рис. 2.10. Результаты расчета ИРЭМ для ВФ Гаусса:

а – зависимость координат экстремумов ИРЭМ;

б – вид ИРЭМ при $k=0.5$.

Исследования показали, что ВФ функция Гаусса имеет более сильные сглаживающие свойства, что не позволяет изменять степень подчеркивания при изменении детальности анализа.

На основе установленной зависимости ИРЭМ предложена методика выбора масштабного уровня ВП при локализации ТО, которая заключается в следующем:

1. Проводится ВП исходного изображения по строкам (либо столбцам).
2. Вычисляется функция ИРЭМ ВП и проводится ее анализ:
 - по положению первого максимума ИРЭМ определяется требуемый масштабный уровень ВП ВФ для локализации отдельных символов ТО;
 - по положению последней точки перегиба ИРЭМ определяется требуемый масштабный уровень ВП ВФ для локализации ТО.

Предложенные математические модели и методика позволят обосновано выбирать вид ВФ и автоматически определять требуемый масштаб, обеспечив целенаправленный выбор параметров обработки при

нахождении пространственных свойств, как отдельных объектов, так и областей при интерпретации изображения с целью получения его абстрактного описания (см. рис. 1.6).

2.4. Вероятностная модель представления символа текста

Для учета пространственных связей между пикселями в изображении символа предложено рассматривать изображение символа как двумерный массив пикселей, в котором определенным образом (в зависимости от шрифта, размеров и др.) расположены информативные (ИП) и фоновые (ФП) пиксели. При этом изображение текстовой области представляет собой упорядоченный определенным образом набор изображений символов.

Формально изображение символа может быть представлено вероятностной моделью символа в виде кортежа [14]

$$S = \{\mathbf{H}, \mathbf{Wh}, \mathbf{P}\}, \quad (2.3)$$

где \mathbf{H} – вектор возможных значений высоты h_i i -го текстового символа (в пикселях);

\mathbf{Wh} – вектор возможных значений ширины wh_i i -го текстового символа (в пикселях);

\mathbf{P} – множество матриц \mathbf{p} , соответствующих вероятности появления ИП в изображениях символа текста размерами $h_i \times wh_i$, $h_i \in \mathbf{H}$, $wh_i \in \mathbf{Wh}$.

Для получения вероятностной модели представления символа текста проведен анализ бинарных изображений текстовых символов, включающих английские буквы ('a', 'b',... , 'z') и десятичные цифры ('0', '1',... , '9') различных типов (Arial, Times New Roman, Calibri, Cambria, Courier New, Adobe Garamond Pro и т.п.) и размеров (14, 15 ..., 20) шрифтов. В результате анализа определены вероятностные характеристики области изображения символа, на основании которых и разработана вероятностная модель .

2.4.1. Характеристики области символа каждой группы символов

Предложено изображения всех символов разбить на 4 группы в зависимости от характера заполнения информативными ипкселями предложенной формы (рис. 2.11, а), которая содержит три зоны (I_1 , I_2 , I_3).

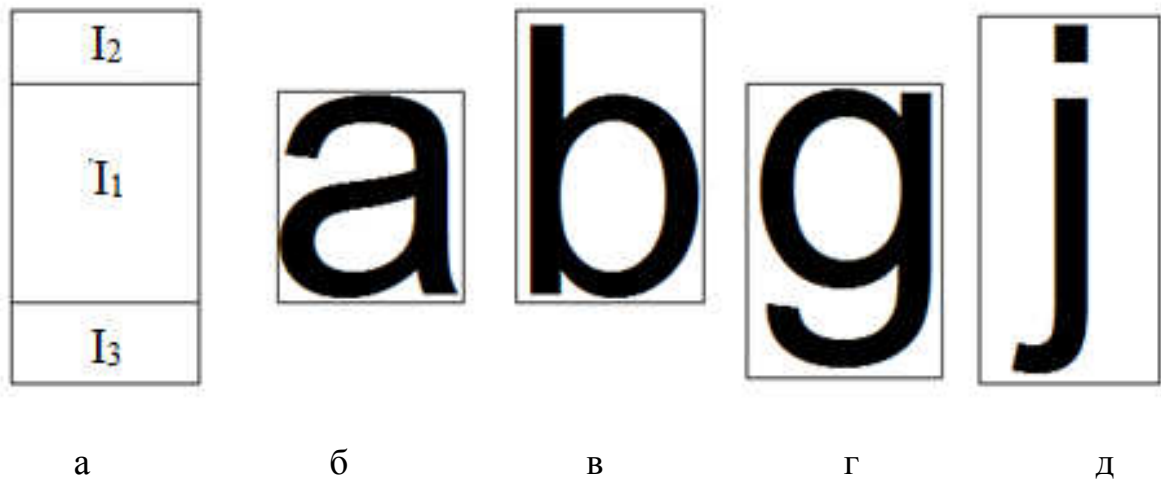


Рис. 2.11. Разбиение символов на группы:

а – форма для разбиения изображения на группы; б, в, г, д – изображения различных символов.

В первой группе содержатся символы букв 'a', 'c', 'e', 'm', 'n', 'o', 'r', 's', 'u', 'v', 'w', 'x', 'z', информативные пиксели которых расположены в зоне I_1 .

Во второй – 'b', 'd', 'f', 'h', 'i', 'k', 'l', 't', 'o', '1', '2', '3', '4', '5', '6', '7', '8', '9', ИП расположены одновременно в зонах I_1 и I_2 .

В третьей – 'g', 'p', 'q', 'r', ИП занимают зоны I_1 и I_3 .

В четвёртой – 'j', ИП занимают все три зоны.

Для каждой группы символов определены параметры модели (2.3).

Полученные кортежи составляют базу вероятностных моделей символов и могут быть использованы для создания изображений псевдосимволов при формировании обучающей выборки.

Для получения матрицы \mathbf{p} из множества \mathbf{P} для каждой группы символов определяется вероятность $p_{i,j} = L_{i,j}/N_{gr}$, появления ИП в каждой точке

изображения где $L_{i,j}$ – количество ИП во всех изображениях символов группы, занимающих позицию с координатами (i, j), N_{gr} – количество символов в группе. Пример матрицы вероятности появления ИП в каждой точке изображения (\mathbf{p}) для символов первой группы приведен в табл. 2.2.

Таблица 2.2 – Матрица \mathbf{p} вероятности появления ИП пикселя в каждой точке изображений символов первой группы размера 13×9 пикселей

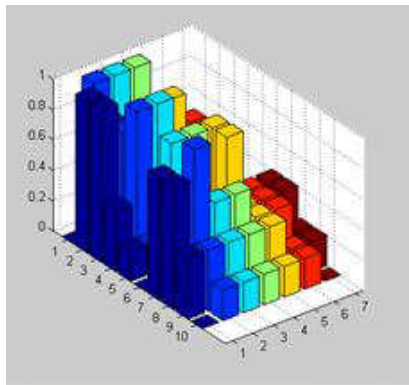
0,194	0,300	0,377	0,467	0,550	0,358	0,141	0,030	0,000
0,203	0,609	0,505	0,386	0,436	0,332	0,201	0,030	0,000
0,256	0,415	0,259	0,274	0,368	0,273	0,274	0,077	0,000
0,252	0,415	0,259	0,274	0,368	0,272	0,274	0,077	0,000
0,223	0,403	0,308	0,329	0,358	0,371	0,236	0,076	0,000
0,188	0,352	0,349	0,259	0,455	0,339	0,277	0,087	0,017
0,209	0,335	0,247	0,279	0,370	0,306	0,285	0,111	0,000
0,242	0,332	0,271	0,294	0,444	0,226	0,296	0,162	0,017
0,209	0,288	0,244	0,292	0,385	0,192	0,238	0,173	0,017
0,223	0,320	0,189	0,265	0,402	0,341	0,242	0,133	0,000
0,227	0,509	0,415	0,406	0,471	0,506	0,409	0,206	0,017
0,139	0,379	0,365	0,289	0,342	0,430	0,311	0,147	0,000
0,120	0,149	0,168	0,117	0,126	0,192	0,123	0,106	0,050

В таблице 2.3 представлено сравнение характеристик групп символов с размером шрифта 14. Например, в первой группе содержатся изображения символов 13-ти букв и цифр с высотой $h = 10$ пикселей. Матрица значений ширины первой группы $\mathbf{Wh} = [7 \ 8 \ 9 \ 10 \ 14]$ и матрица вероятности появления ИП $\mathbf{P} = [0,540 \ 0,400 \ 0,493 \ 0,475 \ 0,503 \ 0,433 \ 0,412 \ 0,529 \ 0,469 \ 0,332 \ 0,392 \ 0,436 \ 0,466]$.

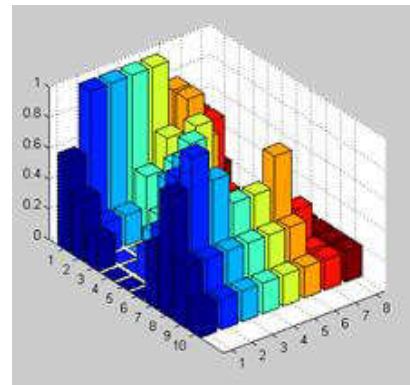
Для изображений символов первой группы визуальные представления матриц \mathbf{p} вероятности появления ИП в соответствующей точке (пикселе) изображения символа размеров 10×7 , 10×8 , 10×9 , 10×10 пикселей представлены на рис.2.12.

Таблица 2.3 – Сравнение характеристики групп символов с размером шифра 14

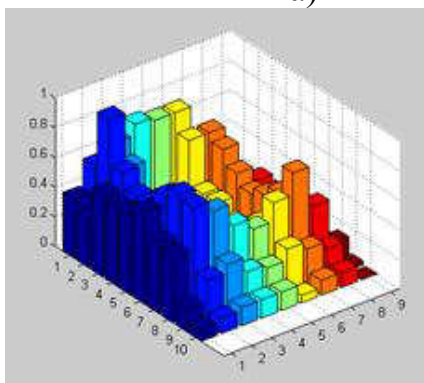
Группа	Символы букв	Высота группы H_{group}	Ширина группы W_{group}	Диапазон вероятности появления ИП в каждой группе символов P_{group}
1	'a', 'c', 'e', 'm', 'n', 'o', 'r', 's', 'u', 'v', 'w', 'x', 'z'	10	[7 8 9 10 14]	[0.332 0.540]
2	'b', 'd', 'f', 'h', 'i', 'k', 'l', 't', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9'	13	[7 8 9 10]	[0.297 0.599]
3	'g', 'p', 'q', 'i'	14	[9 10]	[0.314 0.479]
4	'j'	17	6	0.395



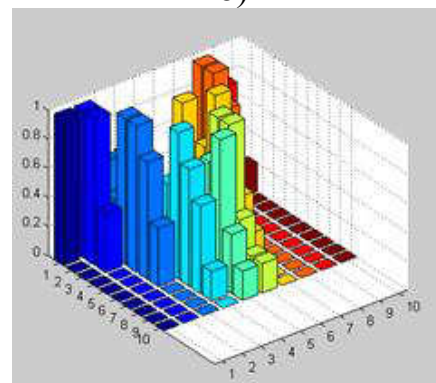
а)



б)



в)



г)

Рис. 2.12. Визуальные представления матриц \mathbf{p} вероятности появления ИП в соответствующей клетке (пикселе) изображения символа размеров:

а – 10x7 пикселей; б – 10x8 пикселей; в – 10x9 пикселей; г – 10x10 пикселей.

2.4.2. Характеристики областей изображения символов букв и цифр

Для каждой буквы и каждой цифры выбираются несколько изображений буквы (цифры) различных типов шрифтов. В каждой точке изображения находится количество появления ИП и количество появления ФП, также вероятности появления ИП на изображении символа текста.

Характеристики области символа (ОС) каждых 36-х букв и цифр получаются следующие – название, ширина, высота, отношение между шириной и высотой, вероятность появления ИП в изображениях буквы (цифра), и матрица вероятности появления ИП в соответствующей точке.

Пример – буква «а». Её ширина – 9; высота – 10; отношение между шириной и высотой – 0,913; вероятность появления ИП в изображениях буквы – 0,540. Визуальные представления матрицы вероятности появления ИП в соответствующей точке (пикселе) представлены на рис. 2.13.

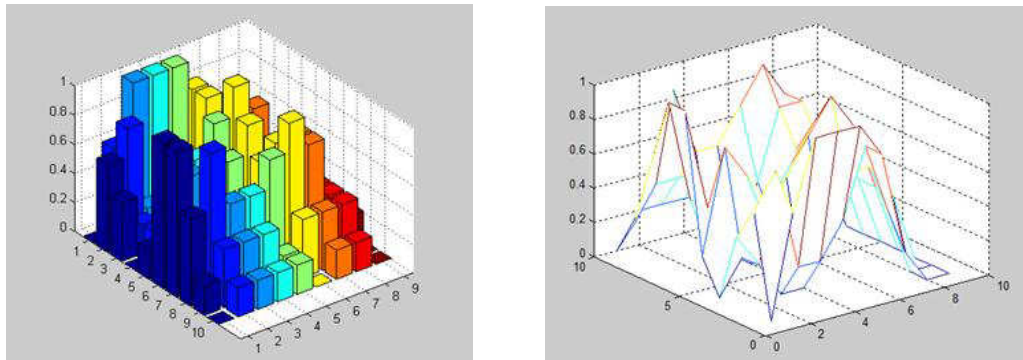


Рис. 2.13. Визуальные представления матрицы вероятности появления ИП в соответствующей клетке (пикселе) буквы «а».

Характеристикой области изображения символа является диапазон вероятности появления ИП в изображениях символов различных типов шрифтов.

Для шрифтов размера 14 условный диапазон вероятности появления ИП в изображениях символов [0,297...0,599]; визуальное представление матрицы вероятности появления ИП в изображениях символов размеров 17x14 пикселей представлены в рис. 2.14.

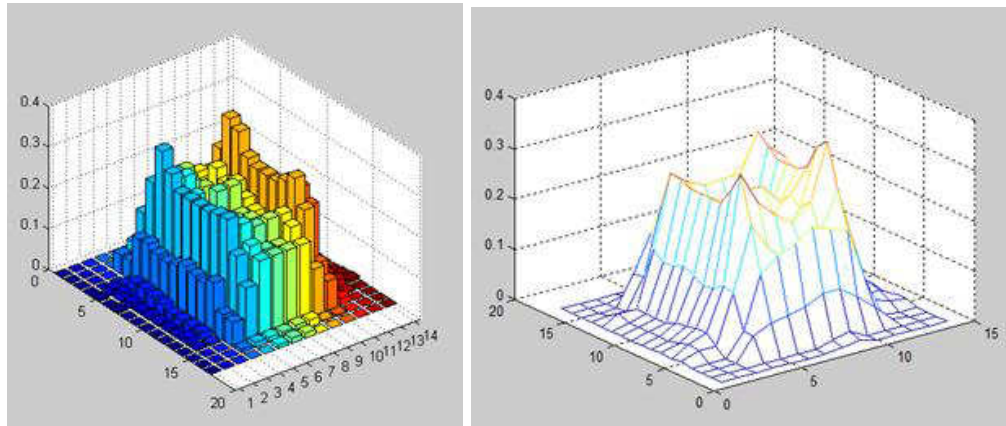


Рис. 2.14. Визуальное представление матрицы вероятности появления ИП в изображениях символов размеров 17x14 пикселей.

В результате проведенных расчетов определены характеристики областей изображения символов различных размеров (от 14 до 20) и типов шрифтов. Как оказалось, диапазон вероятности появления ИП в изображении символа составляет 0,296...0,603 и не зависит от размера шрифта.

Для подтверждения инвариантности вероятностной модели представления символов (2.3) к масштабу изображения показано, что вероятность заполнения изображения символа информативным пикселем $p_{zs} = N_i / (h \cdot wh)$, где N_i – общее количество информационных пикселей в изображении символа, для каждой группы лежит в определенном диапазоне и не зависит от размера шрифта:

- группа 1 – [0,330...0,543];
- группа 2 – [0,296...0,606];
- группа 3 – [0,314...0,482];
- группа 4 – [0,393...0,397].

Проведена оценка одномерного и двумерного подходов к построению моделей ТО. Показано, что использование одномерной модели для решения задачи локализации позволяет автоматизировать процесс обнаружения текстовых строк за счет адаптивного выбора масштаба вейвлет-преобразования. Однако при этом абсолютно не учитываются

пространственные связи между строками, что затрудняет локализацию ТО, особенно на сложном фоне.

Этого недостатка лишено использование для решения задачи локализации вероятностной модели представления символов, которая и была выбрана в качестве базовой для обучения классификаторов.

2.5. Метод формирования текстовой области на основе применения вероятностной модели символов

В процессе формирования обучающей выборки разработчики сталкиваются с противоречием: большой объем обучающей выборки, позволяющий учесть максимальное количество реальных вариантов ТО, обеспечивает улучшение работы классификатора, но значительно увеличивает время обучения, что часто является недопустимым. Кроме того подбор таких баз, как правило, определяется опытом разработчика, т. е. является субъективным.

Поэтому предложено в рамках проводимых исследований по разработке метода локализации текстовых областей такие ТО формировать автоматически с использованием генератора ТО и вероятностной модели символа.

Предложен метод формирования ТО на изображении на основе применения вероятностной модели представления символа, реализованный следующим алгоритмом.

Шаг 1. Выбирается номера группы ($i = 1, \dots, 4$) и инициализация размеров псевдосимвола (высоты h_i , и ширины wh_i) в соответствии с имеющейся базой вероятностных моделей представления символов. Значение высоты h_i выбирается из множества $\mathbf{H} = [h_1, h_2, h_3, h_4] = [10, 13, 14, 17]$, соответствующего четырем группам символов. Для каждого варианта h_i выбираются варианты значения ширины wh_i , соответствующие ширине символов в каждой группе:

$h_1=10$; $wh_1 = [7, 8, 9, 10, 14]$; $\{h_1, wh_1\}=\{(10,7), (10,8), (10,9), (10,10), (10,11)\}$;

$h_2=13$; $wh_2 = [7, 8, 9, 10]$; $\{h_2, wh_2\}=\{(13,7), (13,8), (13,9), (10,10)\}$;

$h_3=14$; $wh_3 = [9, 10]$; $\{h_3, wh_3\}=\{(14,9), (14,10)\}$;

$h_4=17$; $wh_4 = [6]$; $\{h_4, wh_4\}=\{(17,6)\}$.

Шаг 2. Генерируется матрица случайных чисел **TRrand** размера $h_i \times wh_i$, распределенных по равномерному закону в диапазоне $[0 \dots p_{\max}]$, где p_{\max} – максимальное значение вероятности появления ИП в изображениях символов i -й группы.

Шаг 3. Формируется матрица **SM** бинарного изображения псевдосимвола путем сравнения матриц **TRrand** и **p** из базы вероятностных моделей представления символов по следующему правилу:

IF **TRrand** (i, j) \geq **p** (i, j), THEN **SM** (i, j) = ИП, ELSE **SM** (i, j) = ФП.

Шаг 4. Добавляется изображение псевдосимвола к фрагменту изображения текстовой области.

Шаг 5. Повторяются шаги 1-4 до достижения необходимого размера текстовой области на изображении.

Бинарное изображение псевдосимвола получают в результате визуализации полученной матрицы.

В случае необходимости формирования полутонового изображения псевдосимвола, полученное бинарное изображение обрабатывают фильтром Гаусса. Если необходимо иметь цветное изображение псевдосимвола в цветовом пространстве RGB, то полутоновое изображение принимают за R-составляющую, а составляющие G и B получают поэлементным умножением элемента r_{ij} R-составляющей на величину $(1+\varepsilon_{ij})$, где ε_{ij} – случайные числа, распределенные по равномерному закону в диапазоне $[-0.2 \dots 0.2]$.

Изображение текстовой области может быть сгенерировано как упорядоченный определенным образом набор изображений символов. Пример сгенерированного изображения ТО представлен на рис. 2.15.

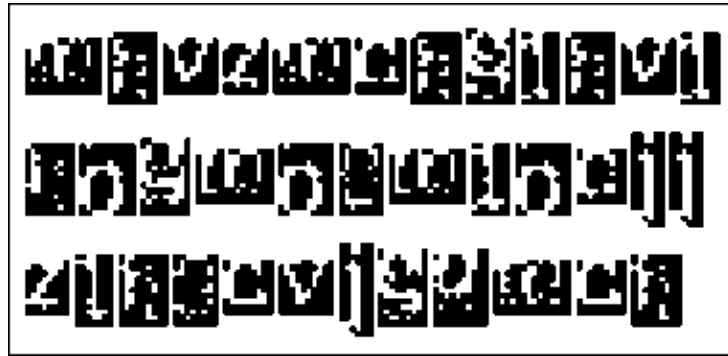


Рис. 2.15. Пример сформированного изображения ТО.

Как было показано в разделе 1, для подчеркивания характерных признаков (контуров) ТО и отдельных символов целесообразно применять ДВП. Особенности его использования согласно обобщенной схеме рис. 2.1 будут рассмотрены в разделе 3.

2.6. Выводы по разделу 2

1. Предложена обобщенная схема ИТ локализации ТО на изображениях на основе двухуровневого подхода, которая включает модель представления ТО и предусматривает два уровня обработки: принятие классификационного решения по результатам корреляционно-экстремального анализа в области вейвлет-преобразования и верификацию ТО-кандидатов.

2. Исследованы модели, используемые в системах переработки графической информации: модель строки символов и модель изображения символа. Показаны особенности поведения модели строки символов в виде квазипериодической последовательности в области вейвлет-преобразования.

3. Усовершенствована вероятностная модель представления символа текста и предложен метод формирования изображений текстовых областей для создания представительной обучающей выборки для обучения классификаторов. Использование вероятностной модели дало возможность формировать изображение псевдосимволов текста с вероятностными характеристиками, соответствующими реальным символам. Это позволило

автоматизировать процесс создания представительной выборки, устранило субъективное влияние на обучение различных классификаторов для выявления текстовых областей на изображениях.

Показано, что использование предложенного метода формирования текстовых областей на изображениях с применением вероятностной модели представления символа текста при формировании обучающей выборки за счет автоматизации процесса создания обучающей выборки существенно уменьшило время её формирования (в среднем на два порядка), что повысило оперативность обнаружения и локализации текстовых областей на изображениях.

3. Предложено использовать вероятностную модель представления символов и метод формирования изображений текстовых областей для обучающей выборки для разработки корреляционно-экстремального метода обнаружения текстовых областей на изображениях на основе двумерного дискретного вейвлет-преобразования.

РАЗДЕЛ 3

РАЗРАБОТКА И ИССЛЕДОВАНИЕ МЕТОДОВ И ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ ЛОКАЛИЗАЦИИ ТЕКСТОВЫХ ОБЛАСТЕЙ НА ИЗОБРАЖЕНИЯХ СО СЛОЖНЫМ ФОНОМ

В третьем разделе получил дальнейшее развитие корреляционно-экстремальный метод обнаружения текстовых областей на изображениях. Метод предусматривает принятие классификационного решения по результатам корреляционно-экстремального анализа в области вейвлет-преобразования согласно обобщенной схеме ИТ локализации ТО на изображениях на основе двухуровневого подхода (см. рис. 2.1).

3.1. Корреляционно-экстремальный метод обнаружения текстовых областей на изображениях

Для получения ТО-кандидатов предложен усовершенствованный корреляционно-экстремальный метод обнаружения ТО на основе вероятностной модели представления символа и двумерного дискретного вейвлет-преобразования (см. рис. 1.8). Метод реализуется следующей последовательностью действий.

Шаг 1. Для обеспечения инвариантности к масштабу и учета как интегральных, так и дифференциальных характеристик символов ТО предложено использовать ДВП-представление изображения – многомасштабное разложение по вейвлет-базису (применяется дискретное вейвлет-преобразование на два уровня) эталонного изображения текстовой области $ET(x,y)$ размером $M \times N$ пикселей

$$\begin{aligned}
 WET_{kp}^A(j_0, m, n) &= \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} ET_{k-1}(x, y) \varphi_{j_0, m, n}(x, y), \\
 WET_{kp}^i(j, m, n) &= \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} ET_{k-1}(x, y) \psi_{j, m, n}^i(x, y), \quad i = \{H, V, D\},
 \end{aligned} \tag{3.1}$$

где $k = 1, 2$ – текущий номер эталонного изображения в разложении, $ET_0(x,y) = ET(x,y)$, $ET_k(x,y) = WET_{k\varphi}^A(j_0, m, n)$ для $k = 1, 2$;

$j_0 = 0, 1$ – исходный масштаб;

$WET_{k\varphi}^A(j_0, m, n)$ – аппроксимирующие коэффициенты;

$WET_{k\psi}^i(j, m, n)$ – детализирующие коэффициенты (горизонтальные H, вертикальные V, диагональные D) для масштаба $j > j_0$, $n = 0, 1, \dots, N-1$, $m = 0, 1, \dots, M-1$ – координаты сдвига вейвлетов $\varphi_{j_0, m, n}$ та $\psi_{j, m, n}^i$.

Шаг 2. Производится двумерное дискретное вейвлет-преобразование на два уровня разложения для фрагмента $FR(x,y)$ входного изображения такого же размера, что и эталонное (получаем $FR_{k\varphi}^A(j_0, m, n)$, $FR_{k\psi}^i(j, m, n)$, аналогично шагу 1).

Шаг 3. Для полученного на каждом уровне разложения фрагмента входного изображения определяется коэффициент корреляции с соответствующими эталонами, полученными на шаге 1

$$KOR_{k\alpha}^i(j, m, n) = WET_{k\alpha}^i(j, m, n) * FR_{k\alpha}^i(j, m, n), \quad \alpha = \{\varphi, \psi\}, \quad i = \{A, H, V, D\}$$

где * – символ операции корреляции.

Шаг 4. Принимается классификационное решение об обнаружении ТО на основании следующего решающего правила

$$\text{IF } \exists KOR_{k\alpha}^i(j, m, n) : (KOR_{k\alpha}^i(j, m, n) > TR), \text{ THEN } FR(x, y) \in TO, \\ \text{ELSE } FR(x, y) \notin TO,$$

где TR – пороговое значение коэффициента корреляции.

3.2. Реализация корреляционно-экстремального метода обнаружения текстовых областей

Проведенные экспериментальные исследования с различными типами вейвлет-функций (Хаара, Добеши, симметричных вейвлетов) не выявили особых преимуществ в использовании определенного типа функции по сравнению с другими, с точки зрения качества обнаружения ТО. Благодаря меньшим вычислительным затратам при реализации вейвлет-преобразования в качестве базисной функции выбрана вейвлет-функция Хаара.

3.2.1. Формирование пирамды изображений для выделенного фрагмента

На данном этапе обработки осуществляется преобразование значений интенсивности фрагмента изображения к диапазону $[-1, 1]$ и выполняется дискретное вейвлет-преобразование с разложением изображения на два уровня по каждой составляющей каналов цвета R, G, B.

Двумерное вейвлет-преобразование представляет собой поочередное одномерное вейвлет-преобразование строк и столбцов матрицы изображения (рис. 3.1.).

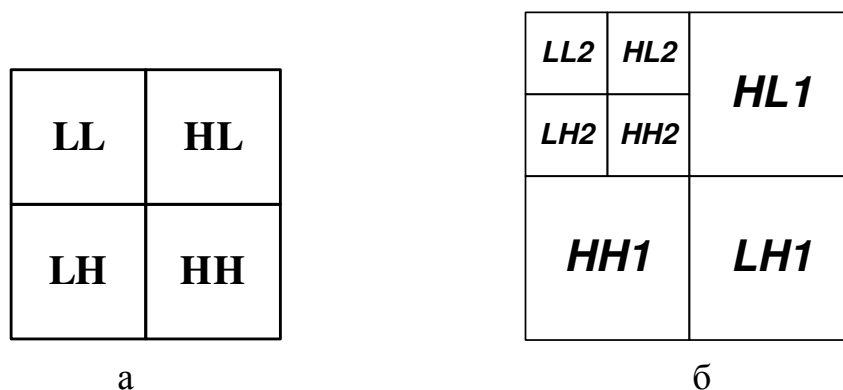


Рис. 3.1. Схема двумерного вейвлет-преобразования изображения: а – однократное; б – двукратное.

Сначала выполняются одномерные вейвлет-преобразования каждой строки, после чего преобразованная строка записывается на прежнее место. Далее вейвлет-преобразования применяются ко всем столбцам. В результате изображение разбивается на четыре равные части. На рис. 3.1 показаны стандартные обозначения квадрантов преобразованного изображения: LL, LH, HL, HH.

Квадрант LL соответствует низкочастотным вейвлет-коэффициентам, HH – высокочастотным. Если не оговорено иное, под N-кратным двумерным вейвлет-преобразованием понимается применение N раз двумерного вейвлет-преобразования, причем очередное двумерное вейвлет-преобразование применяется к младшей четверти матрицы (квадрант LL на рис. 3.1, а).

Полученное N-кратное преобразование показано на рис. 3.1, б ($N=2$), где приведены стандартные обозначения квадрантов изображения. Квадранты N-кратного двумерного вейвлет-преобразования обозначаются аналогично. Указанное преобразование является иерархическим. Иными словами, если при использовании обратного вейвлет-преобразования вычисляются не все уровни, а меньшее их количество, то в квадранте LL образуется уменьшенная копия изображения (рис. 3.2).

Если обратное вейвлет-преобразование не используется, то младший квадрант также является уменьшенной копией изображения. В силу этого свойства, обратное вейвлет-преобразование позволяет вырезать фрагменты изображений различного масштаба. Однако заметим, что, во-первых, доступные масштабы определяются количеством уровней вейвлет-преобразования, и, во-вторых, масштабы не произвольны, а увеличены в два раза.

Схема формирования пирамиды изображений на основе ДВП представлена на рис. 3.3.

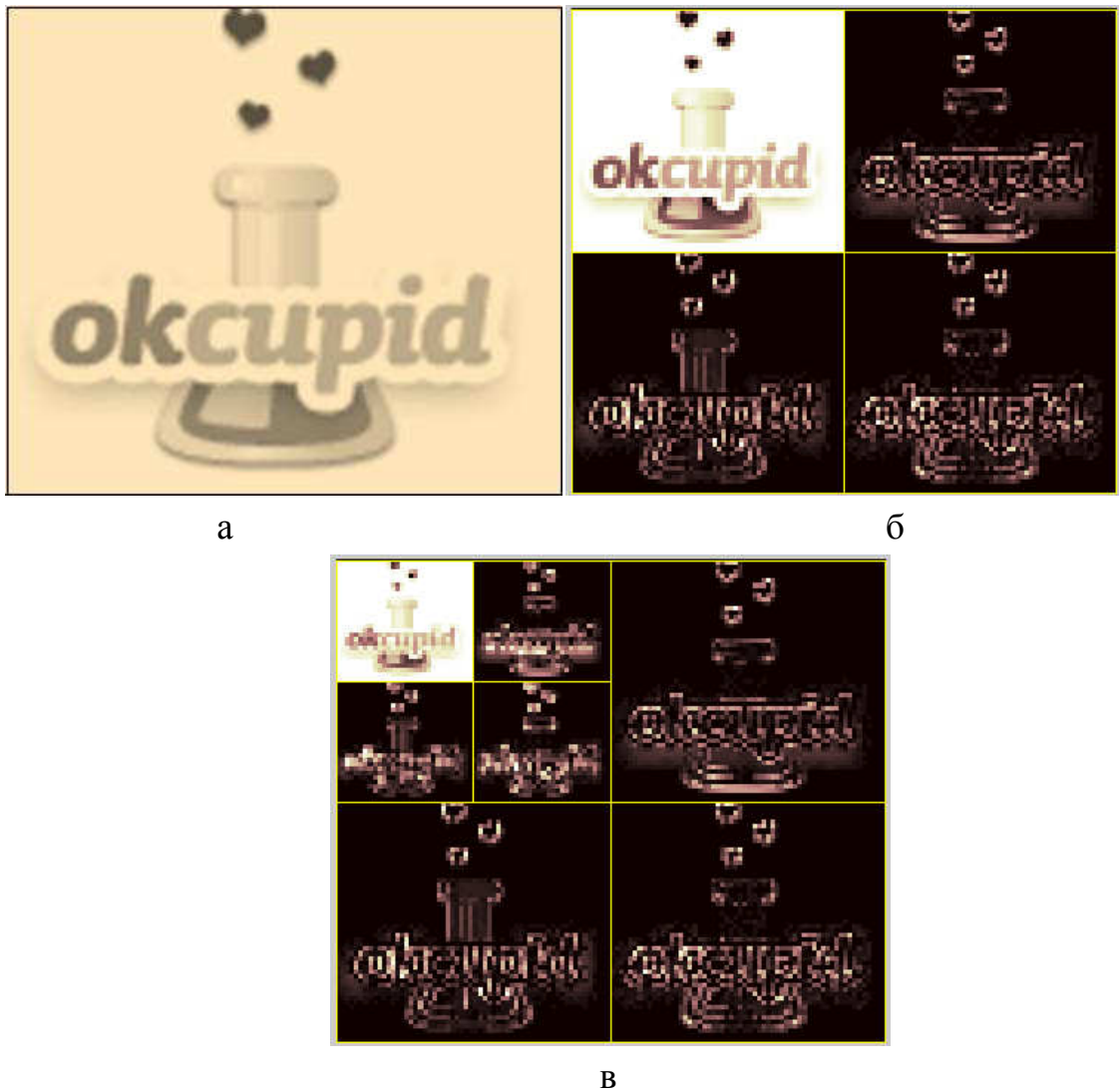


Рис. 3.2. Пример разложения изображения с применением дискретного вейвлет-преобразования Хаара: а – исходное изображение, б – результат однократного ДВП; в – результат двукратного ДВП.

Проведенные исследования показали, что предложенный корреляционно-экстремальный метод обнаружения текстовых областей имеет высокое качество обнаружения ТО, но низкую оперативность, поскольку для каждого фрагмента входного полутонового изображения нужно вычислить $9 \times Nm$ коэффициентов, а для цветного $27 \times Nm$, где Nm – количество эталонных изображений. Для повышения оперативности данного метода предложено применить нейронные сети, позволяющие реализовать процесс параллельных вычислений функций корреляции.

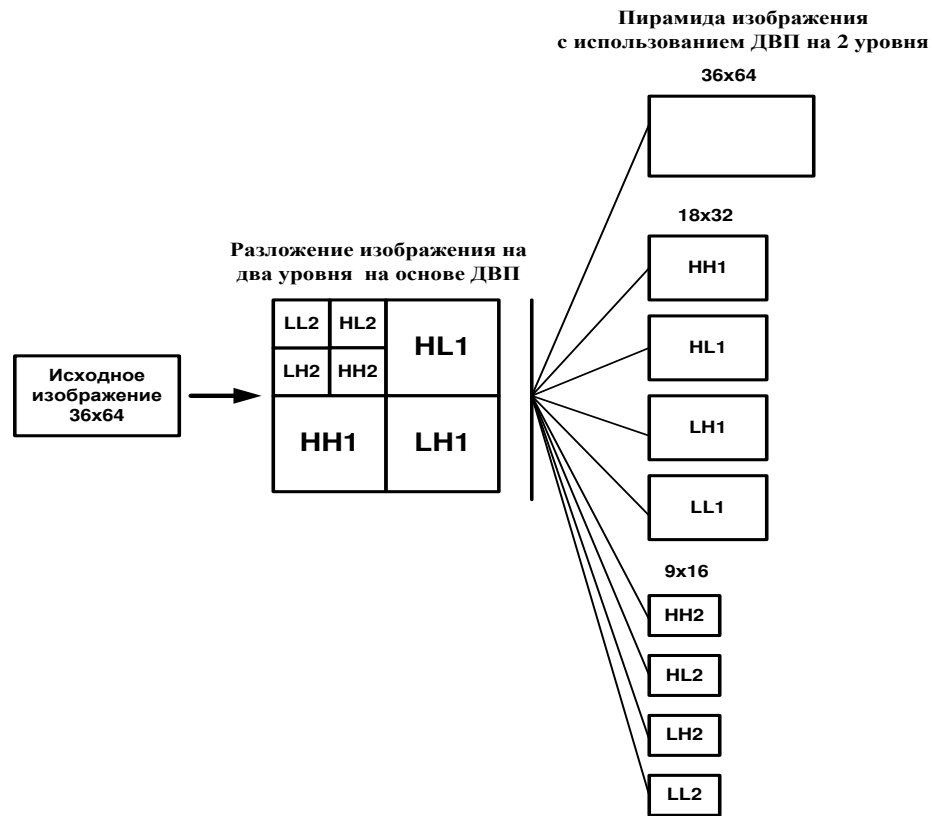


Рис. 3.3. Схема формирования пирамиды изображений на основе ДВП.

3.2.2. Разработка структуры свёрточной нейронной сети для локализации текстовых областей на полутоновых изображениях со сложным фоном

В работе предложена структура свёрточной нейронной сети для обнаружения текстовых областей на полутоновом изображении (рис. 3.4).

Исходные данные для работы сети формируются из полутоновых изображений размером 36×64 пикселя. Сеть классифицирует фрагмент изображения на два класса: «текстовая область» и «не текстовая область».

Свёрточная нейронная сеть состоит из слоя ДВП-представления фрагмента входного изображения, двух свёрточных слоев $S1$ и $S2$ (для их создания используются рецептивные матрицы 5×5 и 3×3 пикселя соответственно), двух слоев подвыборки $S1$ и $S2$ (размер рецептивной матрицы 2×2) и двух полносвязных слоев $N1$ (4 нейрона) и $N2$ (один нейрон) (табл. 3.1).

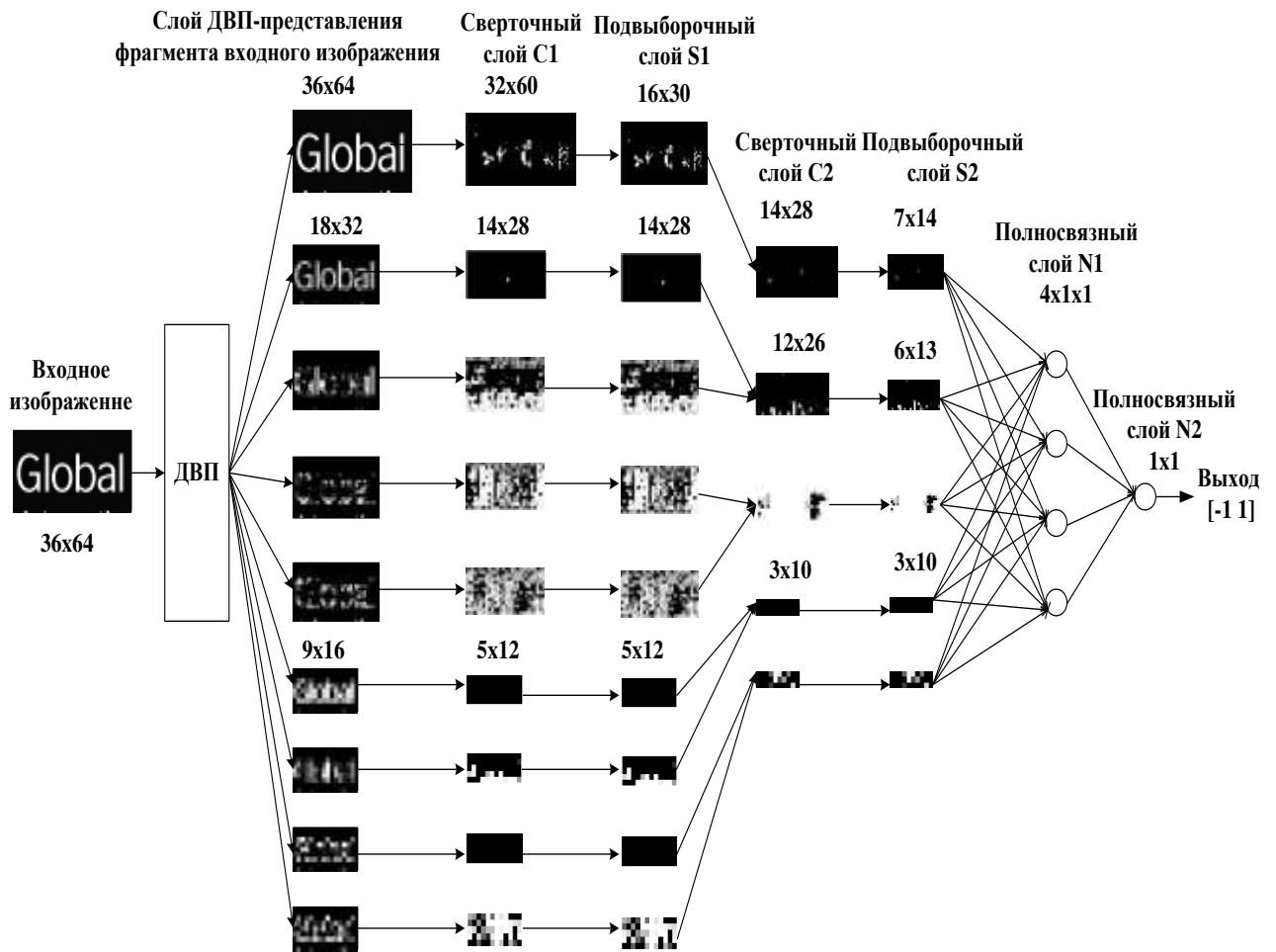


Рис. 3.4. Структура предложенной свёрточной нейронной сети для обнаружения ТО на полутоновых изображениях.

Слой ДВП-представления фрагмента изображения реализует дискретное вейвлет-преобразование в соответствии с выражениями (3.1). На выходе слоя входное изображение представляется одновременно несколькими изображениями различных размеров в виде пирамиды входных изображений (сформировано 9 изображений – два набора по 4 изображения с размерами 18×32 и 9×16 пикселей и входное изображение). Значения интенсивности всех изображений переводятся в диапазон $[-1, 1]$.

Слой $C1$ имеет 9 свёрточных карт, каждая из которых является результатом обработки пирамиды входных изображений с помощью свёртки с маской 5×5 . В качестве функции активации нейронов слоя $C1$ используется гиперболический тангенс.

Таблица 3.1 – Параметры СНС для локализации ТО на полутоновых изображениях со сложным фоном

Название параметра	Входное изображение	Слой C1	Слой S1	Слой C2	Слой S2	Слой N1	Слой N2
Размер, пикселей	36×64	32×60 14×28 5×12	16×30 14×28 5×12	14×28 12×26 3×10	7×14 6×13 3×10	1×1	1×1
Рецептивная карта, пикселей	–	5×5	2×2	3×3	2×2	–	–
Количество карт	–	9	9	5	5	4	1
Функция активации	–	Гиперболический тангенс (ГТ)	Усреднение	ГТ	Усреднение	ГТ	ГТ

Значение элемента карты вычисляется по формуле

$$X_i^{h,l} = f \left(\sum_{k=1}^{n_l-1} \sum_{j=-\infty}^{\infty} X_{i-j}^{k,l} W_{i-j,i}^{h,k,l} + B_i^{h,l} \right), \quad (3.2)$$

где $X_i^{h,l}$ – значение элемента i в карте признаков h слоя l ;

n_l – количество карт признаков в слое l ;

$B_i^{h,l}$ – значение сдвига (смещения) для элемента i в карте признаков h слоя l ;

$W_{i-j,i}^{h,k,l}$ – синаптический вес связи между элементом i в карте признаков h слоя l и элементом $i - j$ карты k слоя $l - 1$.

Особенностью слоя подвыборки S1 является то, что он уменьшает в 2 раза только одну карту слоя C1 размером 32×60 пикселей, другие остаются неизменными. Слой C2 содержит 5 свёрточных карт, каждая из которых получена с помощью свёртки с маской 3×3 некоторых карт слоя S1.

Связь карт слоев S1 и C2, используемая при проектировании данной сети, приведена в табл. 3.2.

Таблица 3.2 – Параметры соединения слоя S1 и слоя C2

№	1	2	3	4	5	6	7	8	9
1	x								
2		x	x						
3				x	x				
4						x	x		
5								x	X

В слое подвыборки S2 карты слоя C2 размером 3×10 дублируются, остальные – уменьшаются в 2 раза. Полносвязные слои N1 и N2 содержат стандартные нейроны с функцией активации в виде гиперболического тангенса. Каждый нейрон в слое N1 соединяется только с одной картой слоя подвыборки S2.

3.2.3. Разработка структуры свёрточной нейронной сети для локализации текстовых областей на цветных изображениях со сложным фоном

В случае обработки цветных изображений нейронная сеть имеет аналогичную структуру для каждого канала R, G, B (пространство RGB).

Предложенный корреляционно-экстремальный метод обнаружения для цветного изображения реализуется следующим алгоритмом.

Шаг 1. С входного цветного изображения (цветовое пространство RGB) выделяется прямоугольный фрагмент размером 36×64 пикселя.

Шаг 2. Формируется пирамида изображений для выделенного фрагмента. Для этого осуществляется преобразование значений интенсивности фрагмента изображения в диапазон [-1, 1] и выполняется

дискретное вейвлет-преобразование с разложением изображения на два уровня по каждой составляющей каналов цвета R, G, B.

Шаг 3. Полученный набор пирамид изображений поступает на предварительно обученную свёрточную нейронную сеть, которая классифицирует данный фрагмент изображения на два класса: «текстовая область» и «не текстовая область». Проводится выбор ГО-кандидатов.

Шаг 4. Повторяются шаги алгоритма 1 - 3 (для всего изображения).

Шаг 5. Формируется список координат ГО-кандидатов.

Структура свёрточной сети представлена на рис. 3.5. Исходные данные для работы сети формируются из цветного изображения (формат RGB) размером 36×64 пикселя. Сеть должна определить, содержит данный фрагмент изображения текст или нет.

Для обнаружения текста из входного изображения с применением ДВП Хаара с разложением изображения на два уровня формируются еще два набора по 4 изображения с размерами 18×32 и 9×16 пикселей для каждой составляющей каналов цвета R, G, B. Значения интенсивностей всех изображений по каждому из каналов – R, G, B преобразуются в диапазон изменения $[-1, 1]$.

Свёрточная нейронная сеть состоит из двух свёрточных слоев (C1, C2), двух подвыборочных слоев (S1, S2) и двух полносвязных слоев (N1, N2) (табл. 3.3). Слой C1 имеет 27 свёрточных карт признаков, которые группируются согласно каналу изображения. Каждая карта получается в результате обработки пирамиды входных изображений с помощью свёртки с маской 5×5 . В качестве функции активации нейронов слоя C1 используется гиперболический тангенс.

Рассмотрим процесс формирования этого слоя, поскольку остальные свёрточные слои формируются подобным образом. Каждый элемент карты признаков соединен с областью размером 5×5 на входном изображении, следовательно, имеет 25 обучаемых коэффициентов и обучаемое смещение. Значение элемента карты вычисляется по формуле (3.2).

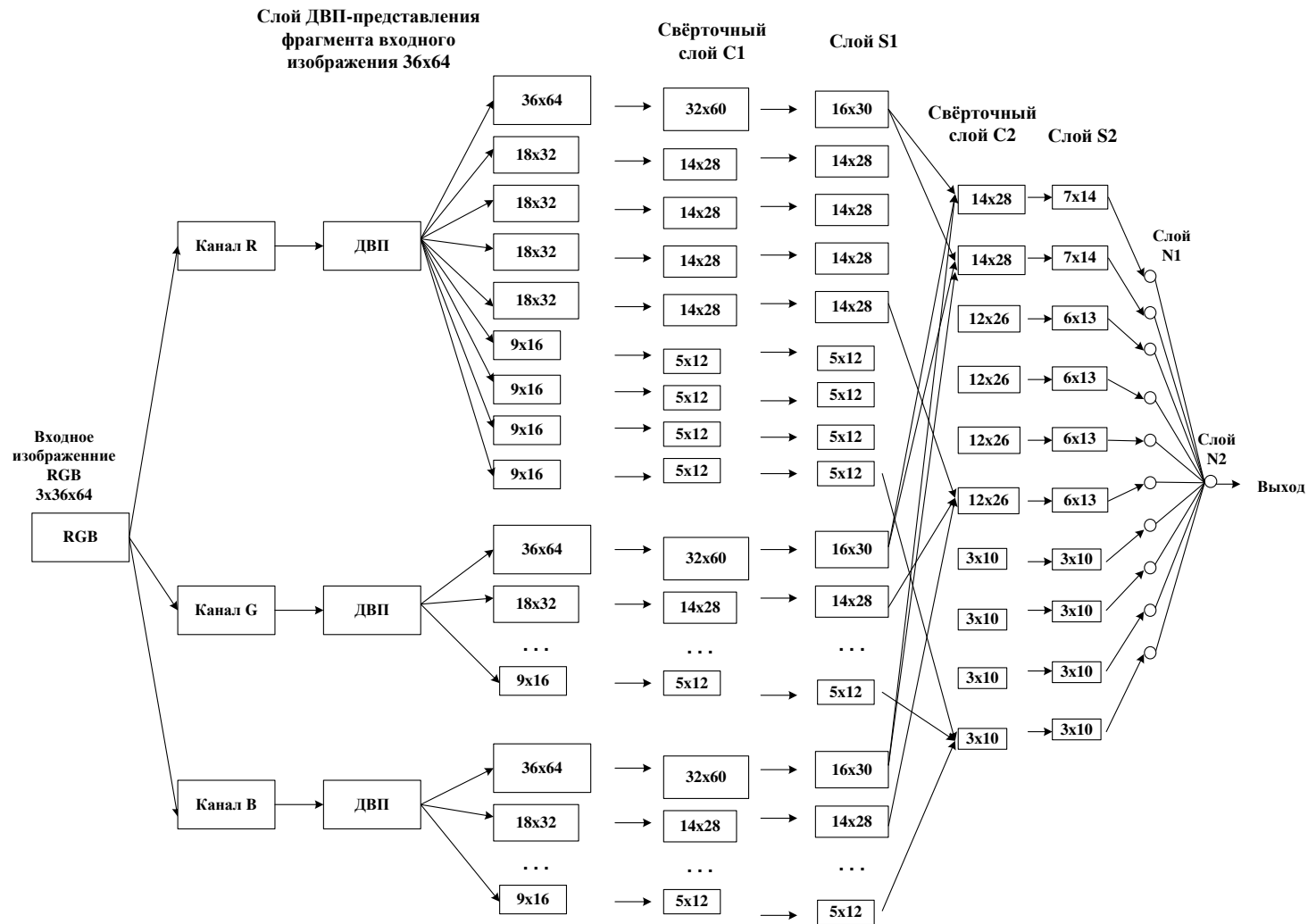


Рис. 3.5. Схема предложенной структуры сети для обнаружения текстовых областей на цветных изображениях со сложным фоном.

Таблица 3.3 – Параметры СНС для локализации ТО на цветных изображениях со сложным фоном

Название параметра	Входное изображение	Слой C1	Слой S1	Слой C2	Слой S2	Слой N1	Слой N2
Размер, пикселей	3×36×64	32×60 14×28 5×12	16×30 7×14 5×12	14×28 5×12 3×10	7×14 5×12 3×10	1×1	1×1
Рецептивная карта, пикселей	–	5×5	2×2	3×3	2×2	–	–
Количество карт	–	27	27	10	10	10	1
Функция активации	–	Гиперболический тангенс (ГТ)	Усреднение	ГТ	Усреднение	ГТ	ГТ

Рецептивные поля соседних элементов на карте признаков соединены с соседними областями предыдущего слоя, следовательно эти поля частично накладываются друг на друга. Как было сказано ранее, все элементы карты признаков имеют общий набор из 25 весов и сдвига, поэтому можно сказать, что они извлекают один и тот же признак во всех возможных областях предыдущего слоя. В данном случае, 27 различных карт второго слоя извлекают 27 различных признаков для разных областей изображения.

Значения всех элементов карты признаков вычисляются путем последовательного прохода по входному слою и применения формулы (3.2) к тем областям, которые являются локальными рецептивными для данных элементов карты признаков.

Интересной особенностью свёрточных слоев является тот факт, что при сдвиге входного изображения значения карт признаков будут сдвинуты на ту же

В подвыборочном слое S2 карты слоя C2 размером 3×10 дублируются, остальные уменьшаются в 2 раза. Наконец, полосвязанные слои N1 и N2 содержат стандартные нейроны. Каждый нейрон в слое N1 соединяется только с одной картой слоя подвыборки S2.

3.3. Алгоритм обучения предложенной свёрточной нейронной сети

Для обучения СНС был выбран алгоритм обратного распространения ошибки с учетом особенностей структуры свёрточной сети. Алгоритм обратного распространения ошибки использует методику, позволяющую быстро вычислять вектор частных производных (градиент) сложной функции многих переменных, если структура этой функции известна. В качестве такой функции в алгоритме рассматривается функция ошибки сети и учитывается тот факт, что структура функции ошибки сети полностью определяется архитектурой нейронной сети, которая считается известной. В процессе обучения достигается настройка межнейронных связей для свёрточных и полносвязных слоев, коэффициенты связей для подвыборочных слоев остаются неизменными и равными 0,25 или 1.

В результате обучения сеть формировала отклики со значениями -1 для фрагмента изображения, не содержащего текст, и $+1$ – для фрагмента изображения с текстом.

Значения исходных синаптических весов для всех свёрточных слоёв сети генерировались в соответствии с законом равномерного распределения с нулевым математическим ожиданием и дисперсией, обратной квадратному корню из количества синаптических связей нейрона [89].

3.3.1. Определение параметров карт слоев свёрточной нейронной сети

Рассмотрим более подробно определение значений весовых коэффициентов карт слоев свёрточной нейронной сети:

а) от входного слоя к свёрточному слою $C1$ (рис.3.6).

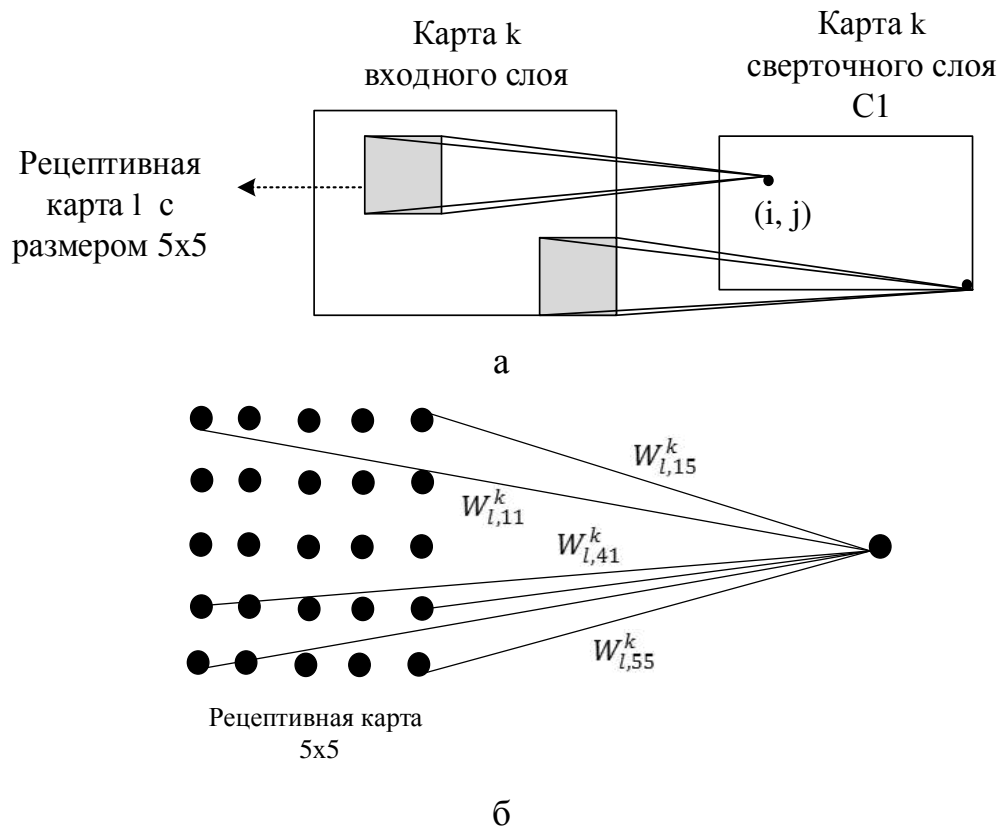


Рис. 3.6. Структура связи элементов входного слоя и свёрточного слоя $C1$:
а – общая схема; б – подробно для элемента (i, j) слоя $C1$.

Синаптический вес связи $W1_l^k$ между элементами рецептивной карты l в карте k входного слоя и элементом в карте k свёрточного слоя $C1$

$$W1_l^k = \begin{pmatrix} W1_{l,11}^k & W1_{l,12}^k & W1_{l,13}^k & W1_{l,14}^k & W1_{l,15}^k \\ W1_{l,21}^k & W1_{l,22}^k & W1_{l,23}^k & W1_{l,24}^k & W1_{l,25}^k \\ W1_{l,31}^k & W1_{l,32}^k & W1_{l,33}^k & W1_{l,34}^k & W1_{l,35}^k \\ W1_{l,41}^k & W1_{l,42}^k & W1_{l,43}^k & W1_{l,44}^k & W1_{l,45}^k \\ W1_{l,51}^k & W1_{l,52}^k & W1_{l,53}^k & W1_{l,54}^k & W1_{l,55}^k \end{pmatrix}.$$

Значение $u1_{i,j}^k$ элемента (i, j) в карте k свёрточного слоя $C1$ вычисляется по формуле

$$u1_{i,j}^k = \frac{e^{S1_{i,j}^k} - e^{-S1_{i,j}^k}}{e^{S1_{i,j}^k} + e^{-S1_{i,j}^k}},$$

где $S1_{i,j}^k = \sum_{i_1=1}^5 \sum_{j_1=1}^5 u0_{i_1,j_1}^k * W1_l^k(i_1, j_1) + B1_{i,j}^k$;

$k = 1, 2, \dots, N$; N – количество карт входного слоя;

$l = 1, 2, \dots, Nl$; $Nl = (x - 4) * (y - 4)$ – количество рецептивных карт карты k входного слоя; (x, y) – размер карты k входного слоя;

$u0_{i_1,j_1}^k$ – значение элемента (i_1, j_1) рецептивной карты l в карте k входного слоя:

$$j = l \bmod (x - 4) + 1; i = (l - j - 1) / (x - 4);$$

$B1_{i,j}^k$ – значение смещения элемента (i, j) в карте k свёрточного слоя $C1$;

б) от слоя $C1$ к слою $S1$ (рис. 3.7).

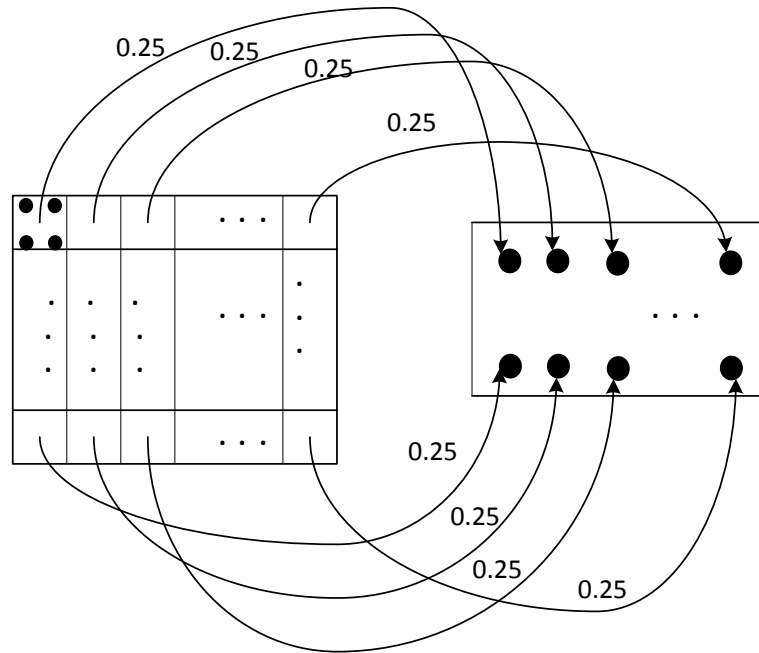


Рис. 3.7. Структура связи элементов свёрточного слоя $C1$ и подвыборочного слоя $S1$.

Синаптический вес связи $W2_l^k$ между элементами рецептивной карты l в карте k свёрточного слоя $C1$ и элементом в карте k подвыборочного слоя $S1$ определяется

$$W2_l^k = \begin{cases} \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix}, & \text{если размер карты слоя } S1 \text{ в 2 раза меньше размера} \\ & \text{карты слоя } C1; \\ \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}, & \text{если размер карт слоёв совпадают.} \end{cases}$$

Если размер карты k подвыборочного слоя $S1$ в 2 раза меньше соответствующей карты свёрточного слоя $C1$, то

$$u2_{i,j}^k = \frac{1}{4} (u1_l^k(i,j) + u1_l^k(i+1,j) + u1_l^k(i,j+1) + u1_l^k(i+1,j+1)),$$

где $k = 1, 2, \dots, N$; N – количество карт свёрточного слоя $C1$;

$l = 1, 2, \dots, N/2$; l – количество рецептивных карт для карты k свёрточного слоя $C1$;

$u1_l^k(i, j)$ – значение элемента (i, j) рецептивной карты l в карте k свёрточного слоя $C1$;

$u2_{i,j}^k$ – значение элемента (i, j) в карте k подвыборочного слоя $S1$;

$i = 1, 2, \dots, \frac{x}{2}$; $j = 1, 2, \dots, \frac{y}{2}$; (x, y) – размер карты k свёрточного слоя $C1$.

Если карта k подвыборочного слоя $S1$ имеет тот же размер, что и карта свёрточного слоя $C1$, то

$$u2_{i,j}^k = u1_{i,j}^k,$$

где $k = 1, 2, \dots, N$; N – количество карт свёрточного слоя $C1$;

$u1_{i,j}^k$ – значение элемента (i, j) в карте k свёрточного слоя $C1$;

$u2_{i,j}^k$ – значение элемента (i, j) в карте k подвыборочного слоя $S1$;

$i = 1, 2, \dots, x$; $j = 1, 2, \dots, y$; (x, y) – размер карты k свёрточного слоя $C1$;

в) от слоя $S1$ к слою $C2$ (рис. 3.8).

Синаптический вес связи $W3_l^{h,k}$ между элементами рецептивной карты l в карте h подвыборочного слоя $S1$ и элементом в карте k свёрточного слоя $C2$ определяется

$$W3_l^{h,k} = \begin{pmatrix} W3_{l,11}^{h,k} & W3_{l,12}^{h,k} & W3_{l,13}^{h,k} \\ W3_{l,21}^{h,k} & W3_{l,22}^{h,k} & W3_{l,23}^{h,k} \\ W3_{l,31}^{h,k} & W3_{l,32}^{h,k} & W3_{l,33}^{h,k} \end{pmatrix}.$$

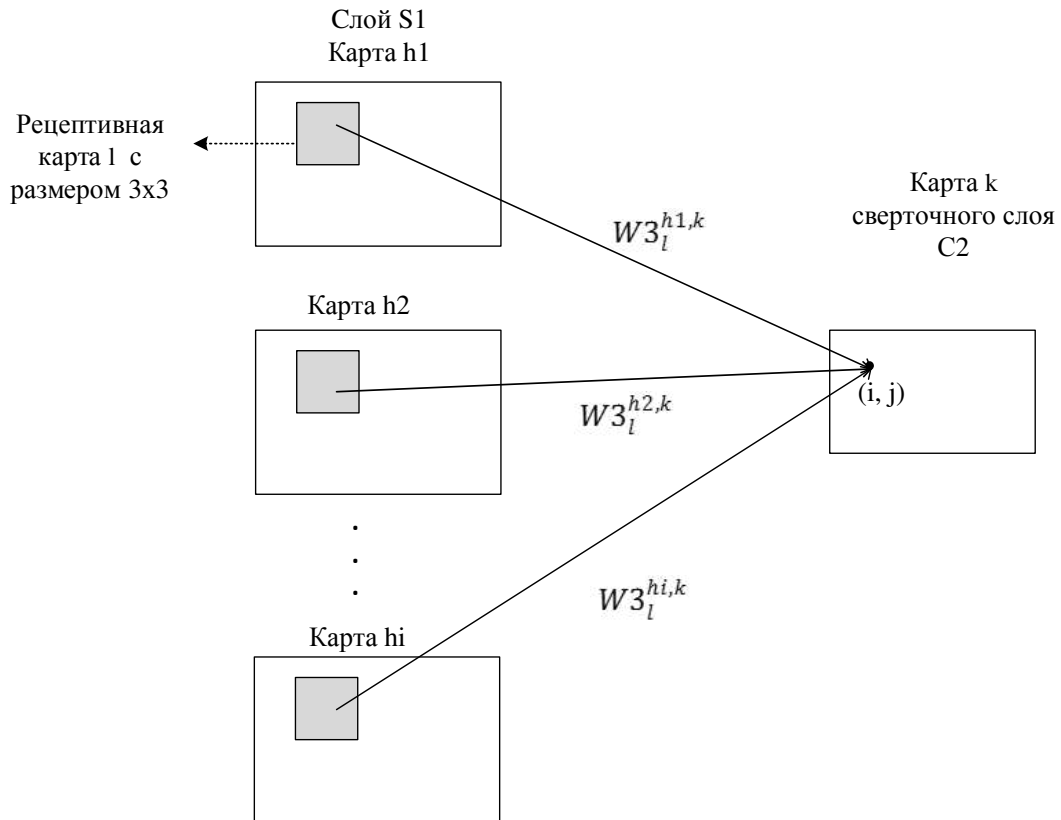


Рис. 3.8. Структура связи элементов подвыборочного слоя S1 и сверточного слоя C2.

Значение $u3_{i,j}^k$ элемента (i, j) в карте k сверточного слоя C2 вычисляется по формуле

$$u3_{i,j}^k = \frac{e^{S3_{i,j}^k} - e^{-S3_{i,j}^k}}{e^{S3_{i,j}^k} + e^{-S3_{i,j}^k}},$$

где $S3_{i,j}^k = \sum_{h=H_k\{n\}}^{nh} \sum_{i_1=1}^5 \sum_{j_1=1}^5 u2_{i_1,j_1}^h * W3_l^{h,k}(i_1, j_1) + B3_{i,j}^k,$

$k = 1, 2, \dots, N$; N – количество карт подвыборочного слоя S1,

$l = 1, 2, \dots, Nl$; $Nl = (x - 2) * (y - 2)$ – количество рецептивных карт карты h подвыборочного слоя S1;

(x, y) – размер карты h подвыборочного слоя S1;

$W3_l^{h,k}$ – коэффициенты связи между элементами рецептивной карты l в карте h подвыборочного слоя S1 и элементом в карте k свёрточного слоя C2;

$u2_l^h(i_1, j_1)$ – значение элемента (i_1, j_1) рецептивной карты l в карте h подвыборочного слоя S1;

$h = H_k\{n\}, n = 1, \dots, nh, H_k$ – множество связи между элементом в карте k свёрточного слоя C2 и картами подвыборочного слоя S1;

nh – количество коэффициентов связи между элементом в карте k свёрточного слоя C2 и картами подвыборочного слоя S1;

$u3_{i,j}^k$ – значение элемента (i, j) в карте k свёрточного слоя C2;

$$j = l \bmod (x - 2) + 1; i = (l - j - 1) / (x - 2);$$

$B3_{i,j}^k$ – значение смещения элемента (i, j) в карте k свёрточного слоя C2;

г) от слоя C2 к слою S2. Синаптический вес связи $W4_l^k$ между элементами рецептивной карты l в карте k свёрточного слоя C2 и элементом в карте k подвыборочного слоя S2

$$W4_l^k = \begin{cases} \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix}, & \text{если размер карты слоя S2 в 2 раза меньше размера} \\ & \text{карты слоя C2;} \\ \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, & \text{если размер карт слоёв совпадают.} \end{cases}$$

Если размер карты k подвыборочного слоя S2 в 2 раза меньше размера карты свёрточного слоя C2, то

$$u4_{i,j}^k = \frac{1}{4} (u3_l^k(i, j) + u3_l^k(i + 1, j) + u3_l^k(i, j + 1) + u3_l^k(i + 1, j + 1)),$$

где $k = 1, 2, \dots, N$; N – количество карт свёрточного слоя C2;

$l = 1, 2, \dots, N/2$; l – количество рецептивных карт карты k свёрточного слоя

$C2$;

$u3_l^k(i, j)$ – значение элемента (i, j) рецептивной карты l в карте k свёрточного слоя $C2$;

$u4_{i,j}^k$ – значение элемента (i, j) в карте k подвыборочного слоя $S2$;

$i = 1, 2, \dots, \frac{x}{2}$; $j = 1, 2, \dots, \frac{y}{2}$; (x, y) – размер карты k свёрточного слоя $C2$.

Если карта k подвыборочного слоя $S2$ имеет тот же размер, что и соответствующая карта свёрточного слоя $C2$, то

$$u4_{i,j}^k = u3_{i,j}^k,$$

где $k = 1, 2, \dots, N$; N – количество карт свёрточного слоя $C2$;

$u3_{i,j}^k$ – значение элемента (i, j) в карте k свёрточного слоя $C2$;

$u4_{i,j}^k$ – значение элемента (i, j) в карте k подвыборочного слоя $S2$;

$i = 1, 2, \dots, x$; $j = 1, 2, \dots, y$; (x, y) – размер карты k свёрточного слоя $C2$;

д) от слоя $S2$ до слоя $N1$ (рис. 3.9).

Значение $u5_k$ элемента (i, j) в карте k свёрточного слоя $C1$ вычисляется по формуле

$$u5_k = \frac{e^{S5_k} - e^{-S5_k}}{e^{S5_k} + e^{-S5_k}}$$

где $S5_k = \sum_{h=1}^{NS2} \sum_{i=1}^x \sum_{j=1}^y u4_h(i, j) * W5_h^k(i, j) + B5_k$

$k = 1, 2, \dots, n1$; $n1$ – количество нейронов полносвязанного слоя $N1$;

$W5_h^k(i, j)$ – синаптический вес связи между элементом (i, j) в карте h подвыборочного слоя $S2$ и нейроном k полносвязанного слоя $N1$;

$h = 1, 2, \dots, NS2$; $NS2$ – количество карт подвыборочного слоя $S2$;

(x, y) – размер карты h подвыборочного слоя S2;

$u_{4h}(i, j)$ – значение элемента (i, j) в карте h подвыборочного слоя S2;

u_{5k} – значение нейрона k полносвязанного слоя N1;

B_{5k} – значение смещения нейрона k полносвязанного слоя N1;

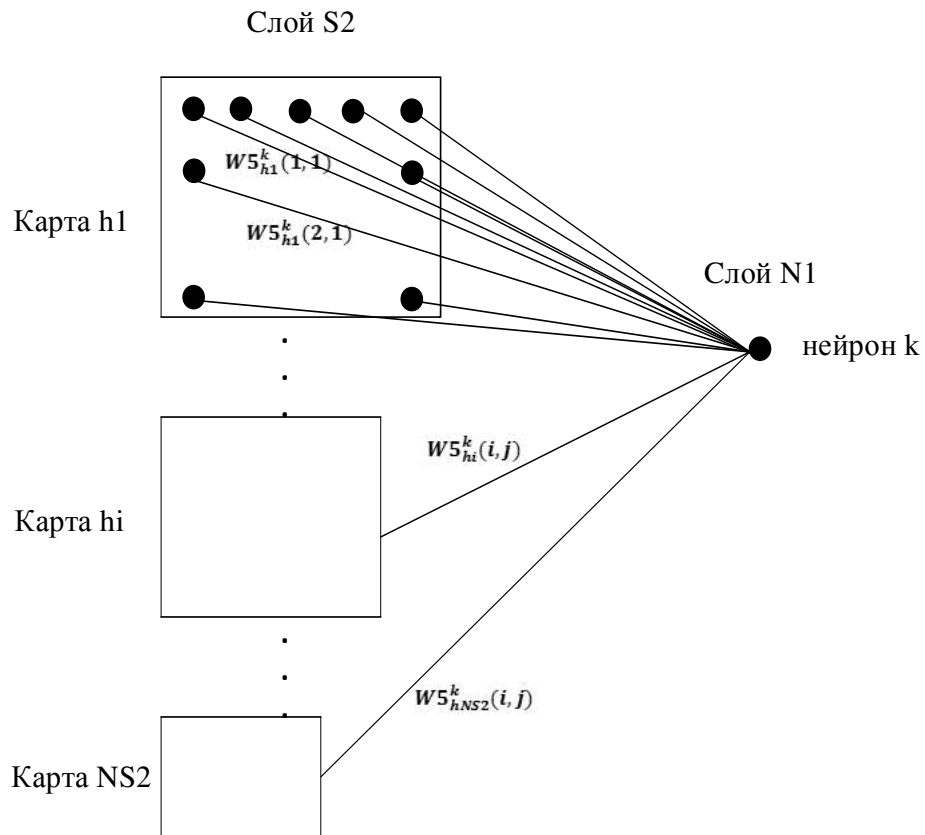


Рис. 3.9. Структура связи элементов подвыборочного слоя S2 и и элемента k полносвязного слоя N1.

е) от N1 к N2 (выходной слой) (рис. 3.10).

Значение u_6 элемента (i, j) в карте k полносвязного слоя N2 вычисляется по формуле

$$u_6 = \frac{e^{S_6} - e^{-S_6}}{e^{-S_6} + e^{-S_6}} ,$$

где $S_6 = \sum_{k=1}^{n_1} W_{6k} * u_{5k} + B_6$,

$k = 1, 2, \dots, n_1$; n_1 – количество нейронов полносвязанного слоя N1;

W_{6k} – синаптический вес связи между нейроном k полносвязанного слоя N1 и нейроном полносвязанного слоя N2;

u_{5k} – значение на выходе нейрона k полносвязанного слоя N1;

u_6 – значение на выходе нейрона полносвязанного слоя N2 (выходное значение сети);

B_6 – значение смещения нейрона k полносвязанного слоя N2.

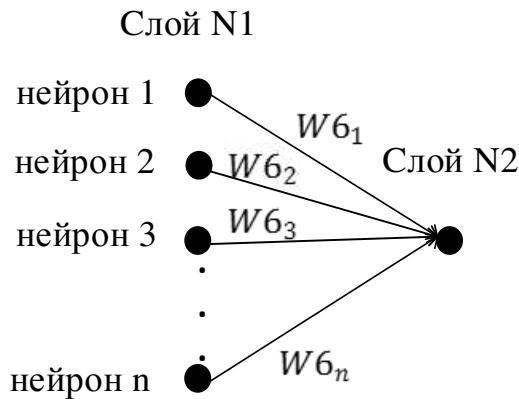


Рис. 3.10. Структура связи элементов полносвязанного слоя N1 и выходного элемента полносвязанного слоя N2.

3.3.2. Определение ошибок свёрточной нейронной сети

Сеть обучается методом обратного распространения исходя из обеспечения минимальной среднеквадратической ошибки классификации

$$E = \frac{1}{2} \sum (C - out)^2 \rightarrow \min,$$

где $C = \{-1, 1\}$ – возможное (целевое) значение на выходе сети (распознается текстовая область – выход +1, не текстовая – выход -1).

Поэтому для обучения сети согласно методу обратного распространения ошибки находим производную ошибки по коэффициенту w с учетом того, что в качестве функции активации нейронов используется гиперболический тангенс:

$$\frac{\partial E}{\partial w} = (C - out) \cdot u_6 \cdot (1 - out^2) \cdot u_w,$$

где $out = u_6$ – выход сети,

u_w – выход нейрона, который связан с выходом сети коэффициентом w .

Ошибки слоев рассчитываются следующим образом:

– ошибка слоя N2

$$err_{N2} = 0,5 \cdot (C - u_6) \cdot (1 - u_6) \cdot (1 + u_6);$$

– ошибка слоя N1

$$err_{N1_k} = 0,5 \cdot (1 + u_{5_k}) \cdot (1 - u_{5_k}) \cdot err_{N2} \cdot W_{6_k},$$

где $k = 1, \dots, n1$; $n1$ – количество нейронов полносвязанного слоя N1;

– ошибка слоя S2

$$err_{S2_h} = 0,5 \cdot (1 + u_{4_h}) \cdot (1 - u_{4_h}) \cdot err_{N1} \cdot W_{5_h},$$

где $h = 1, 2, \dots, NS2$; $NS2$ – количество карт подвыборочного слоя S2;

– ошибка слоя C2

$$err_{C2_h} = err_{S2_h} \cdot W_{4_h},$$

где $h = 1, 2, \dots, NC2$; $NC2$ – количество карт слоя C2;

– ошибка слоя S1

$$err_{S1_h} = 0.5 \cdot (1 + u3_h) \cdot (1 - u3_h) \cdot \sum err_{C2_k} \cdot w3_h^k,$$

где $h = 1, 2, \dots, NS1$; $NS1$ – количество карт слоя S1;

k – номер карт слоя C2, которые связаны с картой h слоя S1;

– ошибка слоя C1

$$err_{C1} = err_{S1_h} \cdot W2_h,$$

где $h = 1, 2, \dots, NC2$; $NC2$ – количество карт слоя C1.

3.3.3 Коррекция весов связи между слоями и значений смещения

В ходе обучения сети для каждого из слоев выполняется коррекция весов:

а) от слоя N1 к слою N2

$$newW6_k = W6_k + \rho * err_{N2} * u5_k,$$

$$newB6 = B6 + \rho * err_{N2};$$

б) от слоя S2 к слою N1

$$newW5_h^k(i, j) = W5_h(i, j) + \rho * err_{N1_k} * u4_h(i, j),$$

$$newB5_k = B5_k + \rho * err_{N1_k};$$

в) от слоя S1 к слою C2

$$newW3_l^{h,k}(i, j) = W3_l^{h,k} + \rho * err_{C2_k}(i, j) * u2_l^h(i, j),$$

$$newB3_k(i, j) = B3_k(i, j) + \rho * err_{C2_k}(i, j);$$

г) от входного слоя к слою $C1$

$$newW1_i^k(i, j) = W1_i^k(i, j) + \rho * err_{C1_k}(i, j) * u0_i^k(i, j),$$

$$newB1_k(i, j) = B1_k(i, j) + \rho * err_{C1_k}(i, j),$$

где $\rho - const, \rho \in (0,1)$.

3.4 Исследование применения свёрточной нейронной сети для обнаружения текстовых областей на изображениях со сложным фоном

Обучающая выборка сформирована с помощью разработанных вероятностной модели представления символа текста и метода формирования текстовых областей на изображениях. Автоматизация процесса создания обучающей выборки позволила устранить влияние субъективных факторов при формировании и повысить уровень ее представительства. В ходе обучения изображения сформированных ТО последовательно подавались на вход свёрточной нейронной сети. В результате обучения сеть формировала отзывы со значениями 1 для фрагмента изображения, который не содержит текст, и 1 – для фрагмента изображения с текстом. Проверка качества обучения сети осуществлялась отдельно на контрольных выборках (по 500 изображений) цветных и полутоновых изображений. Каждая контрольная выборка включала 50% изображений ТО, сформированные с помощью предложенного метода формирования текстовых областей на изображениях, и 50% реальных изображений.

Для оценки качества обнаружения текстовых областей (табл. 3.5) использованы такие оценки, как ошибки первого и второго рода, меры точности Precision (Pr) и полноты Recall (Rc).



Рис. 3.11. Примеры изображений контрольной выборки.

Таблица 3.5 – Результаты оценивания качества обнаружения текстовых областей

Условия формирования обучающей выборки		Pr, %	Rc, %	Ошибка I рода, %	Ошибка II рода, %
без использования вероятностной модели представления символа текста		77,7	80,0	22,3	20,0
с использованием вероятностной модели представления символа	полутоновые изображения	87,7	86,6	12,1	12,9
	цветные изображения	87,3	87,2	12,4	12,7

Таким образом, качество обнаружения текстовых областей контрольной выборки (500 изображений на сложном фоне) с использованием предложенного корреляционно-экстремального метода и его реализации на основе свёрточной нейронной сети из ДВП-представлением изображения на основе дискретного вейвлет-преобразования и учебным набором на основе созданных текстовых областей составила 87,7%, что на 10% выше качества обнаружения ТО сетью без использования учебных наборов на основе созданных текстовых областей (77,7%).

За счет автоматизации процесса создания обучающей выборки существенно уменьшилось время ее формирования (до минут по сравнению с

часами и днями, необходимыми для получения и обработки достаточного количества реальных изображений ТО), что повысило оперативность обнаружения ТО, а использование свёрточной нейронной сети с ДВП-представлением изображения позволило уменьшить зависимость результатов обнаружения ТО от размеров и типов шрифтов текста.

Для обучения сети была создана обучающая выборка из 676 изображений (36×64 пикселя) со сложным фоном и контрольная выборка из 560 изображений. Изображения контрольной выборки включали текст с различными размерами, типами и цветом шрифтов, многострочный текст. Кроме символов текста они содержали и другие объекты (фрагменты домов, деревьев и т.д.). Также были добавлены изображения, содержащие только часть символов текста и без текста.

Типичные графики процесса обучения сети с использованием созданного обучающего набора представлены на рис. 3.12.

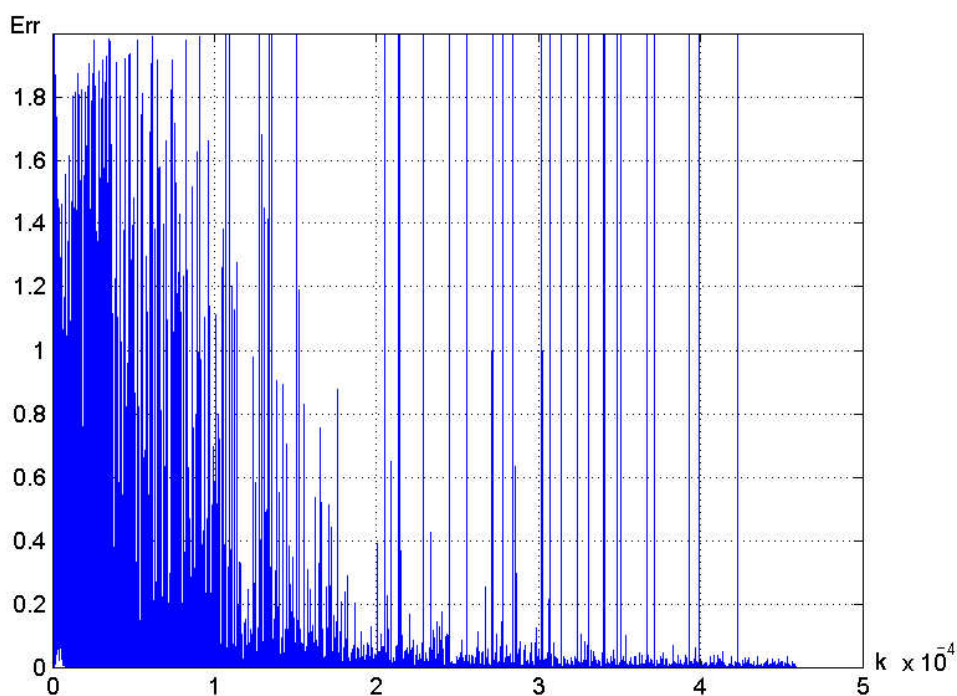


Рис. 3.12. Графики процесса обучения сети, с использованием созданного обучающего набора.

3.5. Метод локализации тестовых областей на изображениях со сложным фоном на основе свёрточной нейронной сети с использованием многомасштабного представления изображения на основе ДВП

Предложено усовершенствовать метод локализации текстовых областей за счет использования разработанного корреляционно-экстремального метода обнаружения ТО и морфологической обработки изображений текстовых областей.

Усовершенствованный метод локализации реализуется следующим алгоритмом:

Шаг 1. ТО-кандидат, полученная с помощью разработанного метода обнаружения ТО, подвергается дискретному двумерному вейвлет-преобразованию с вейвлет-функцией Хаара. В результате формируется пирамида из пяти (для полутоновых) или 15 (для цветных, пространство RGB) изображений.

Шаг 2. Для каждого из изображений, визуализирующих детализирующие коэффициенты, выполняется морфологическая операция – дилатация со структурными элементами размера 3×5 , 3×3 и 7×5 пикселей (для горизонтальных, диагональных и вертикальных изображений детализирующих коэффициентов соответственно).

Шаг 3. Обработанные изображения вместе с входным и двумя изображениями аппроксимирующих коэффициентов подаются на вход свёрточной нейронной сети, которая окончательно классифицирует ТО-кандидат на два класса: «текстовая область» или «не текстовая область».

Шаг 4. Уточняются координаты текстовой области и определяются ее границы в виде отрезков прямых линий.

В результате применения предложенного метода локализации ТО ко всему изображению, определяются координаты прямоугольных областей, содержащих

СИМВОЛЫ ТЕКСТА.

Схема метода локализации ТО на изображениях со сложным фоном на основе СНС с использованием многомасштабного представления изображения на основе ДВП представлена на рис. 3.13.

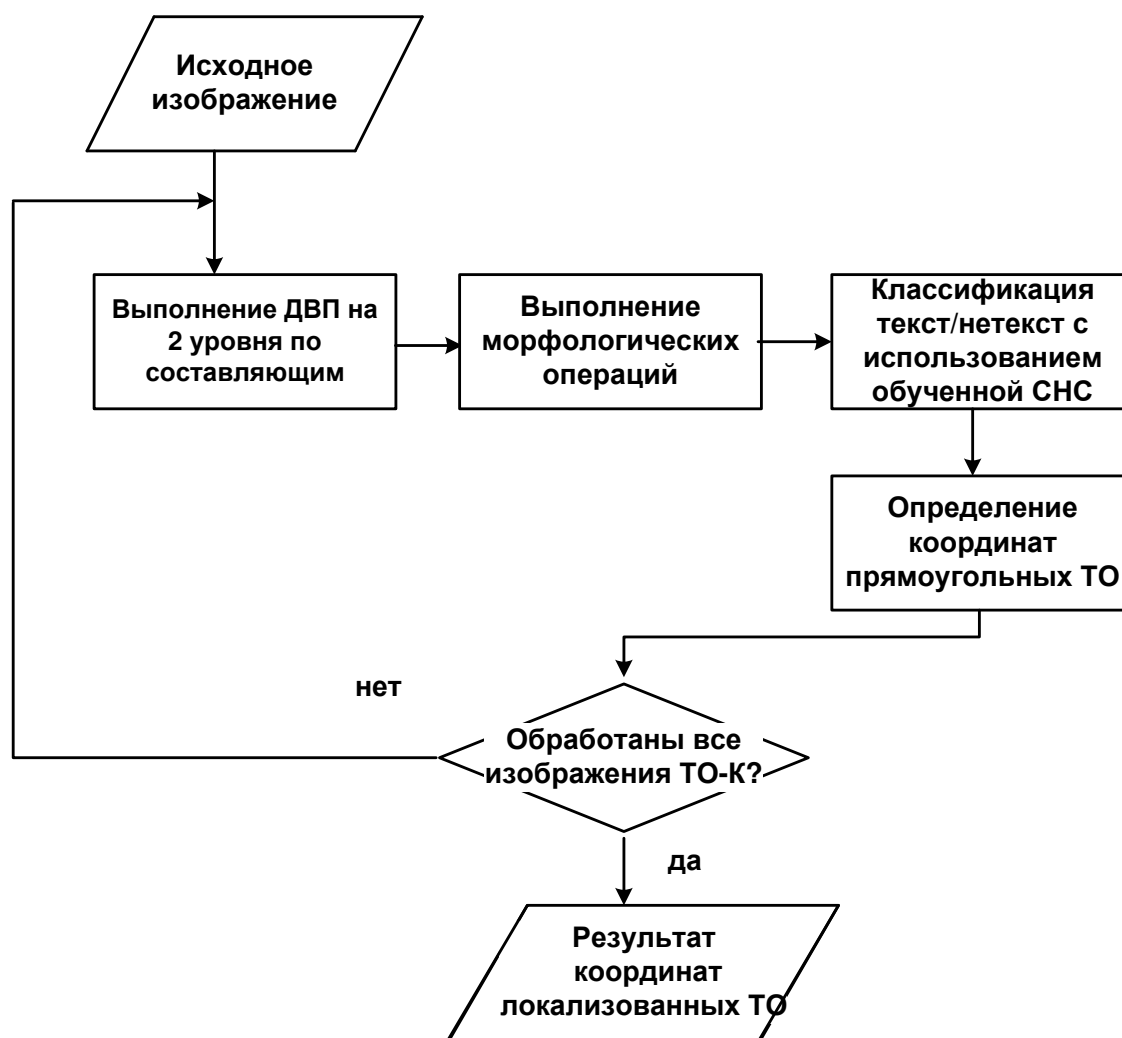


Рис. 3.13. Схема метода локализации ТО на изображениях со сложным фоном на основе СНС с использованием многомасштабного представления изображения на основе ДВП.

Для тестирования работы нейронной сети при локализации ТО

оценивались точность классификации фрагментов изображений из обучающей и контрольной выборок, содержащих текстовые области.

Для сравнения точности локализации текстовых областей выполнено локализацию набора изображений со сложным фоном из телевизионных роликов, фильмов, рекламных баннеров с помощью предложенной ИТ локализации ТО и для сравнения – с применением программ FineReader11 и FreeOCR.

Показателем качества, характеризующим точность локализации, избран метрический показатель близости между множеством координат эталонных границ ТО тестового изображения и множеством координат границ, полученным в результате применения рассматриваемого метода локализации (критерий FOM (Figure of Merit), предложенный Прэттом) в соответствии с формулой (1.5)

Использование предложенного метода локализации позволило повысить точность локализации на 10,5% по сравнению с корреляционно-экстремальным методом обнаружения ТО.

3.6 Выводы по разделу 3.

Таким образом, в разделе разработаны корреляционно-экстремальный метод обнаружения текстовых областей на изображениях, реализованный на базе свёрточной нейронной сети, и метод локализации текстовых областей на основе разработанного корреляционно-экстремального метода обнаружения ТО и морфологической обработки изображений текстовых областей, которые послужили основой для создания информационной технологии локализации текстовых областей на изображениях со сложным фоном для автоматической переработки графической информации.

Качество обнаружения текстовых областей контрольной выборки (500 изображений на сложном фоне) с использованием предложенного корреляционно-экстремального метода и его реализации на основе свёрточной нейронной сети из ДВП-представлением изображения на основе дискретного вейвлет-преобразования и учебным набором на основе созданных текстовых областей составила 87,7%, что на 10% выше качества обнаружения ТО сетью без использования учебных наборов на основе созданных текстовых областей (77,7%).

За счет автоматизации процесса создания обучающей выборки существенно уменьшилось время ее формирования (до минут по сравнению с часами и днями, необходимыми для получения и обработки достаточного количества реальных изображений ТО), что повысило оперативность обнаружения ТО, а использование свёрточной нейронной сети с ДВП-представлением изображения позволило уменьшить зависимость результатов обнаружения ТО от размеров и типов шрифтов текста.

Усовершенствование метода локализации позволило повысить точность локализации на 10,5% по сравнению с корреляционно-экстремальным методом обнаружения ТО.

РАЗДЕЛ 4

РАЗРАБОТКА И ТЕСТИРОВАНИЕ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ ЛОКАЛИЗАЦИИ ТЕКСТОВЫХ ОБЛАСТЕЙ

Информационные технологии переработки графической информации, в которых решается задача локализации текстовых областей на изображениях со сложным фоном, находят широкое применение при решении различных прикладных заданий, таких как бухгалтерский документооборот, охранная деятельность, робототехника, учет транспорта, идентификация продукции, поиск текстовой информации в мультимедийных коллекциях.

В данном разделе на основе разработанных вероятностной модели представления символа текста, метода формирования текстовой области, корреляционно-экстремального метода обнаружения и метода локализации текстовых областей разработана и протестирована информационная технология локализации текстовых областей на изображениях.

4.1. Обобщенная структура информационной технологии локализации текстовых областей на изображении

При разработке обобщенной структуры информационной технологии локализации текстовых областей на изображении учтены:

- предложенная во втором разделе схема ИТ локализации ТО на изображениях на основе двухуровневого подхода (п. 2.1, рис. 2.1);
- разработанные во втором разделе вероятностная модель представления символа текста (п. 2.4) и метод формирования текстовой области (п. 2.5);
- разработанные в третьем разделе корреляционно-экстремальный метод обнаружения текстовых областей на изображениях со сложным фоном (п. 3.1) и свёрточные нейронные сети для его реализации (п. 3.2);

– разработанный метод локализации текстовых областей на изображениях со сложным фоном (п. 3.5).

Предложена обобщенная структура информационной технологии локализации текстовых областей на изображении (табл. 4.1).

Таблица 4.1 – Структура информационной технологии локализации текстовых областей на изображении

Название этапа	Выполняемая процедура	Модель или метод, предложенные для использования
I. Подготовительный (этап обучения)	1.1. Создание обучающей выборки	Вероятностная модель представления символа текста (п. 2.4)
	1.1.1 Создание базы вероятностных моделей представления символов текста	
	1.1.2 Формирование изображения ТО на основе применения вероятностной модели представления символа	Метод формирования изображения ТО на основе применения вероятностной модели представления символа (п. 2.5);
	1.2 Обучение сверточной нейронной сети для обнаружения ТО-кандидатов	Корреляционно-экстремальный метод обнаружения текстовых областей (КЭМО ТО), использующий многомасштабное представление изображения на основе двумерного дискретного вейвлет-преобразования. (п. 3.1, п. 3.2)
1.3 Обучение сверточной нейронной сети для локализации ТО	Усовершенствованный метод локализации текстовых областей (УМЛ ТО) на основе разработанного корреляционно-экстремального метода обнаружения ТО и морфологической обработки изображений текстовых областей. (п. 3.5)	

Продолжение табл. 4.1

Название этапа	Выполняемая процедура	Модель или метод, предложенные для использования
II. Основной (этап – верификация и окончательная локализация ТО)	2.1 Обнаружение ТО-кандидатов	КЭМО ТО (п. 3.2)
	2.2 Выполнение ДВП на 2 уровня по составляющим	УМЛ ТО (п. 3.5)
	2.3 Выполнение морфологических операций	УМЛ ТО (п. 3.5)
	2.3 Классификация текст/нетекст с использованием обученной СНС	УМЛ ТО (п. 3.5)
	2.4 Определение координат локализованных ТО	УМЛ ТО (п. 3.5)

Разработанная информационная технология содержит два этапа.

На первом этапе, подготовительном (этапе обучения), выполняются следующие процедуры:

- создание обучающей выборки, которая представляет собой набор искусственно созданных изображений ТО (бинарных, полутоновых или цветных в зависимости от прикладной задачи) на основе применения вероятностной модели представления символа;

- выбор структур и обучение сверточных нейронных сетей для обнаружения ТО-кандидатов и локализации ТО.

Следует отметить, что в ходе подготовительного этапа происходит обучение двух сверточных нейронных сетей, имеющих подобную структуру. Кроме того, для повышения быстродействия СНС и сокращения их времени обучения для полутоновых изображений следует использовать более упрощенную структуру сети, чем для цветных (см. рис. 3.4., рис. 3.5).

Для обучения сверточной нейронной сети для обнаружения ТО-кандидатов использовалась обучающая выборка, полученная на предыдущем шаге.

Характеристики обучения СНС обнаружения ТО-кандидатов на полутоновых и цветных изображениях представлены ниже (табл. 4.2). В таблице показано, что процент правильного обнаружения искусственно созданных изображений ТО обучающей выборки не ниже 99,6%.

Таблица 4.2 – Характеристики обучения СНС обнаружения ТО-кандидатов с использованием обучающего набора из моделей ТО

Характеристики	СНС обнаружения ТО-кандидатов на полутоновых изображениях	СНС обнаружения ТО-кандидатов на цветных RGB изображениях
Количество изображений обучающей выборки	2310	1836
Количество эпох обучения (K)	34650	45900
Время обучения на эпоху (с)	0.2	0.61
Максимальное значение ошибки после окончания процесса обучения	0.064	0.032
Процент верного обнаружения для обучающей выборки (ПВО), (%)	99.6	99.9

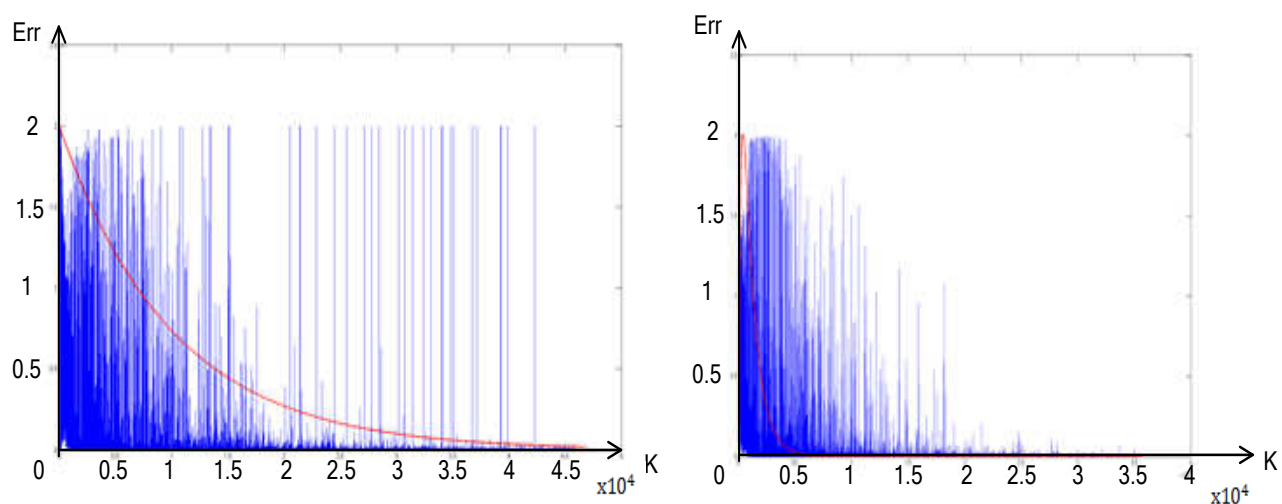
Эмпирические значения ошибки обучения сети E_{tr} в зависимости от количества эпох обучения k изображены в виде гистограммы, а непрерывные линии получены согласно аппроксимирующим выражениям (4.1) для локализации текстов на цветных изображениях в цветовом пространстве RGB, а для полутоновых изображений согласно выражениям (4.2)

$$\text{Err}(k) = 2 * e^{-0.001 * (k - 2.5)}, \quad (4.1)$$

$$\text{Err}(k) = 3.75 * (e^{-0.001 * (k - 0.5)} - e^{-0.005 * (k - 0.5)}), \quad (4.2)$$

где k – количество эпох обучения сети.

На основании полученных выражений можно предварительно оценить время обучения сети при заданной величине ошибки. Графики ошибки обучения СНС обнаружения ТО на цветных и полутоновых изображениях с использованием созданной обучающей выборки приведены на рис. 4.1.



а)

б)

Рис. 4.1 Графики ошибки обучения СНС для обнаружения ТО (а) на цветных изображениях, (б) на полутоновых изображениях.

Особенностью обучения сверточной нейронной сети для локализации ТО является то, что в качестве обучающей выборки используется набор ТО-кандидатов, определенных с помощью СНС для обнаружения ТО.

Схема функционирования информационной технологии локализации ТО на изображении представлена на рис. 4.2. На основном этапе (этапе верификации и окончательной локализации ТО) выполняются процедуры в соответствии с табл. 4.2.

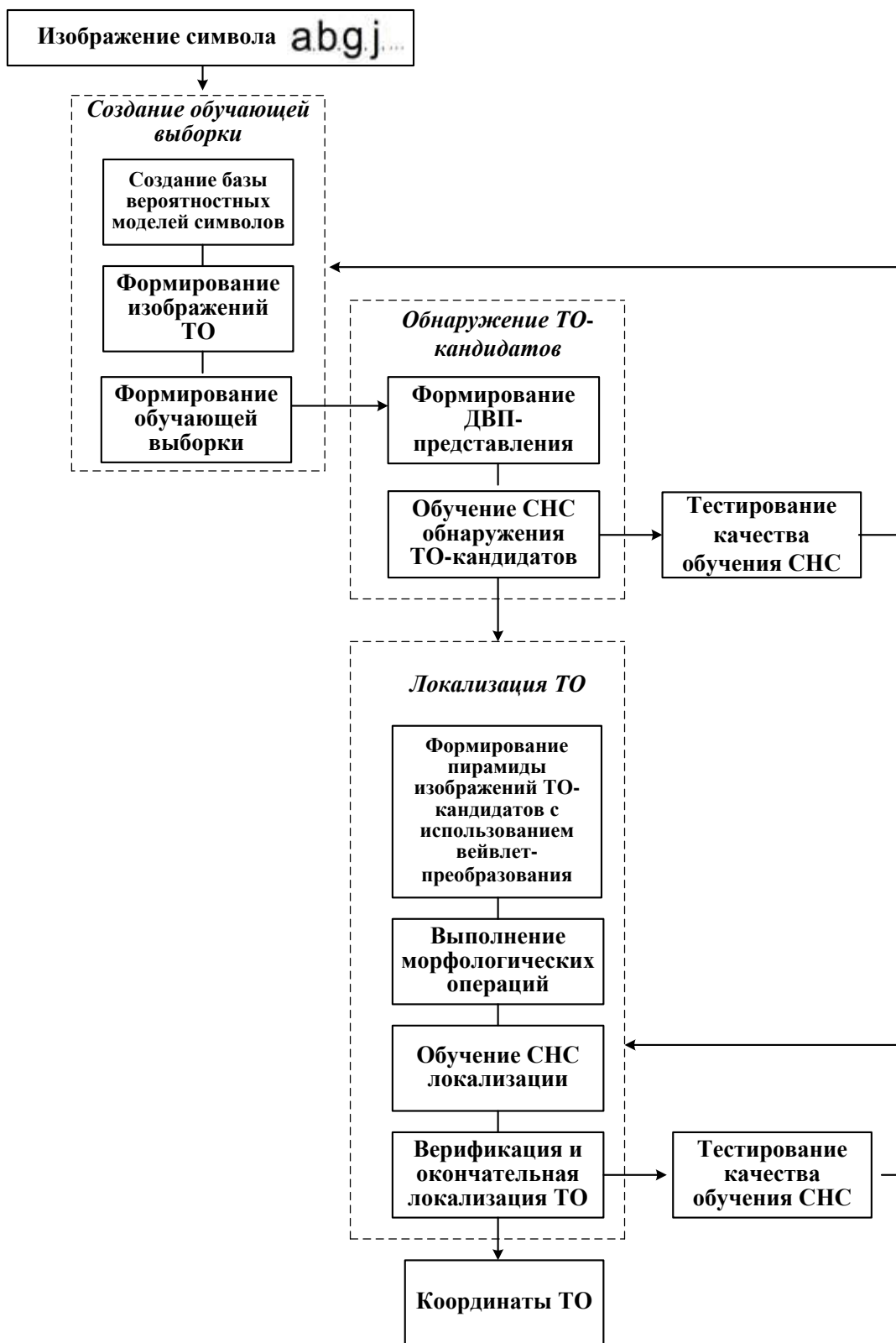


Рис. 4.2. Схема функционирования информационной технологии локализации ТО на изображении на этапе обучения.

На основном этапе используются предварительно обученные сверточные нейронные сети, которые с помощью корреляционно-экстремального метода обнаружения ТО и усовершенствованного метода локализации ТО выполняют окончательную классификацию ТО-кандидата на два класса: «текстовая область» и «не текстовая область».

Подробная схема функционирования информационной технологии локализации ТО на изображении на основном этапе представлена на рис. 4.3.

4.2. Разработка специализированного программного комплекса

Разработка специализированного программного комплекса проводилась в следующих направлениях:

1. Создание специализированной среды, предоставляющей средства для проведения исследовательских работ по обработке изображений.
2. Создание программно-алгоритмического обеспечения для систем локализации ТО.

В специализированном программном комплексе реализованы следующие процедуры, используемые при обработке и анализе изображений, в том числе, и рассмотренные в предыдущих разделах:

- предварительная обработка изображений;
- вейвлет-анализ изображений;
- локализация объектов на изображении;
- выделение контуров и бинаризация изображений;
- анализ контуров;
- статистический анализ обрабатываемых изображений;
- отладка разрабатываемого программного обеспечения;
- оценка эффективности алгоритмов.

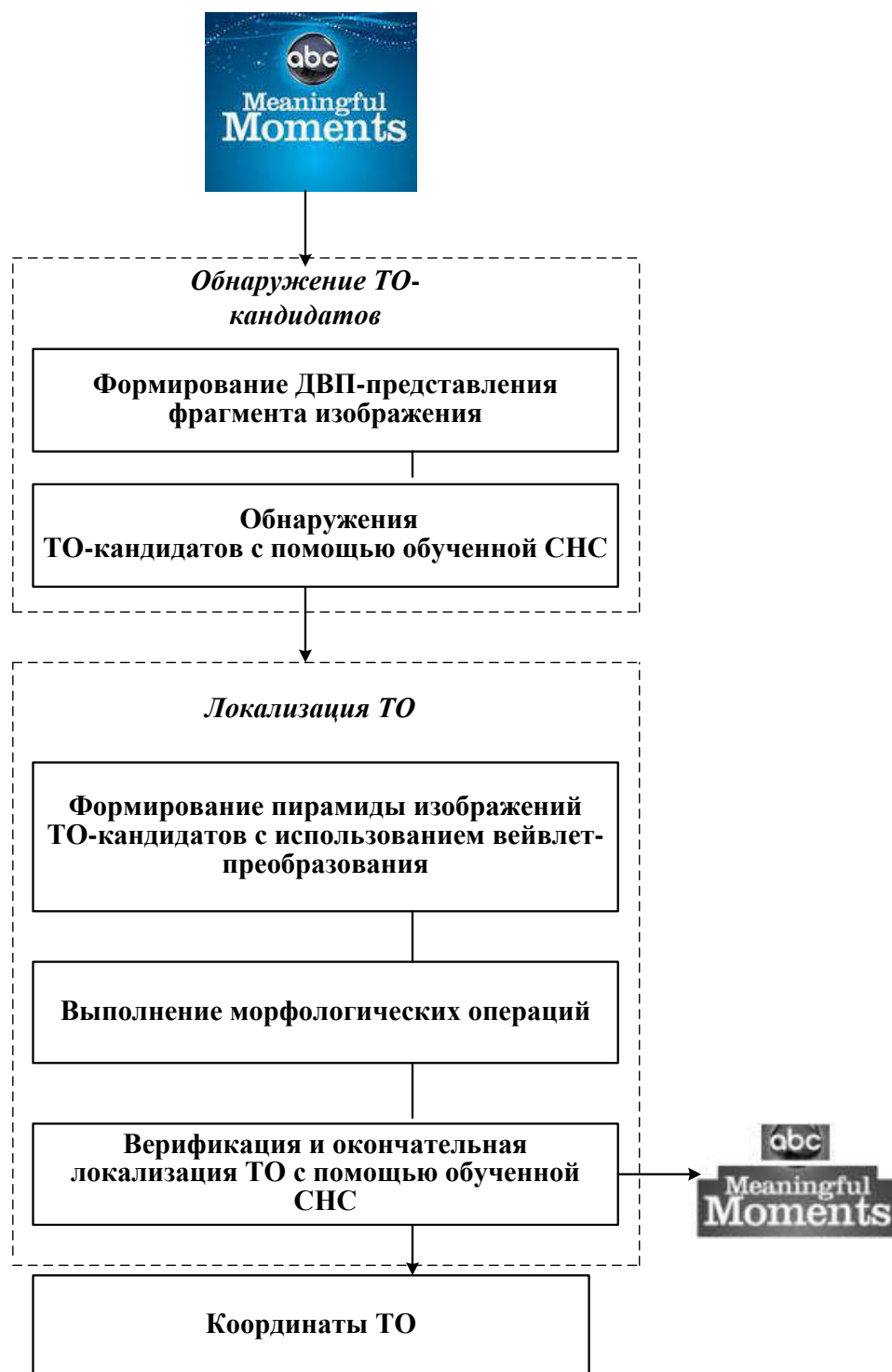


Рис. 4.3. Схема функционирования информационной технологии локализации ТО на изображении на основном этапе.

Специализированный программный комплекс способен поддерживать следующие форматы изображений:

- Bitmap Files (*.bmp);
- JPEG (*.JPG, *.JPEG, *.JPE, *.JFIF);
- GIF (*.GIF);
- TIFF (*.TIFF, *.TIF);
- PNG (*.PNG);

Ниже представлены примеры использования специализированного программного комплекса для проверки в тестовом режиме работоспособности предложенных моделей и методов ИТ локализации ТО на изображении. Некоторые экранные формы представлены на рис. 4.4 – 4.7. Последовательность действий, соответствующих тестовому режиму работы системы, приведена ниже.

1. В режиме отладки работы системы для выбора загружаемого файла следует нажать кнопку “ImageBrowser” (рис. 4.4.).

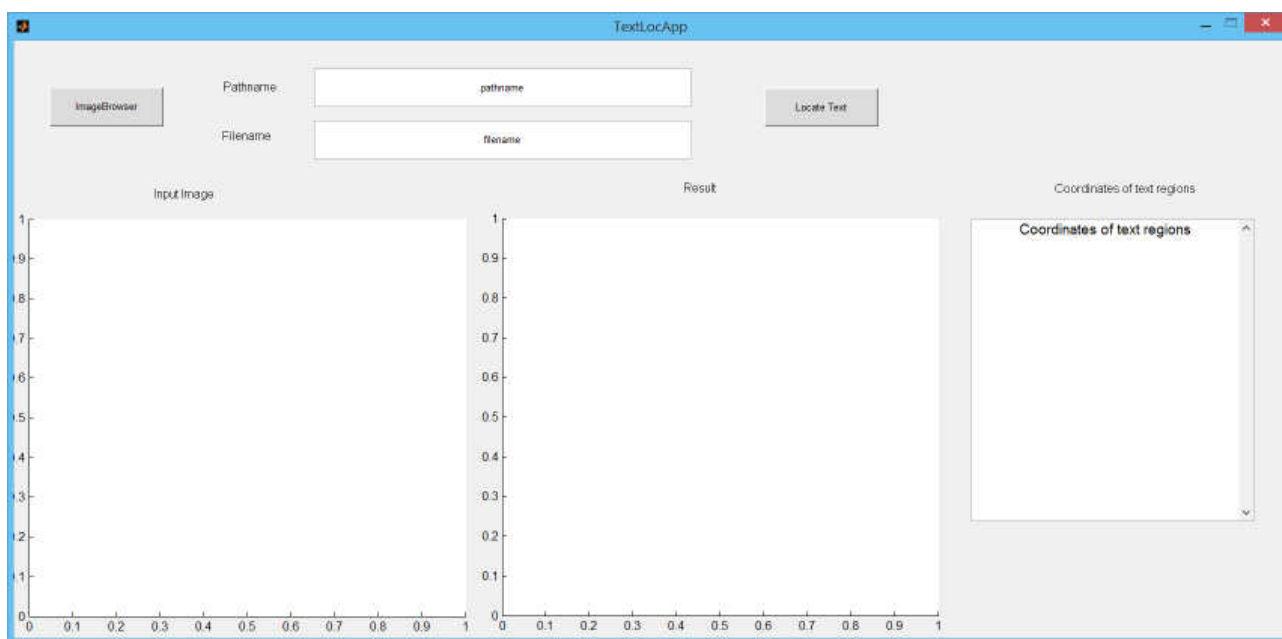


Рис. 4.4. Главное окно программы локализации.

2. В открывшемся окне выбирается файл изображения (рис. 4.5.).

Файл изображения загружается и появляется в главном окне с информацией о файле (рис. 4.6).



Рис. 4.5. Выбор файла с изображением.

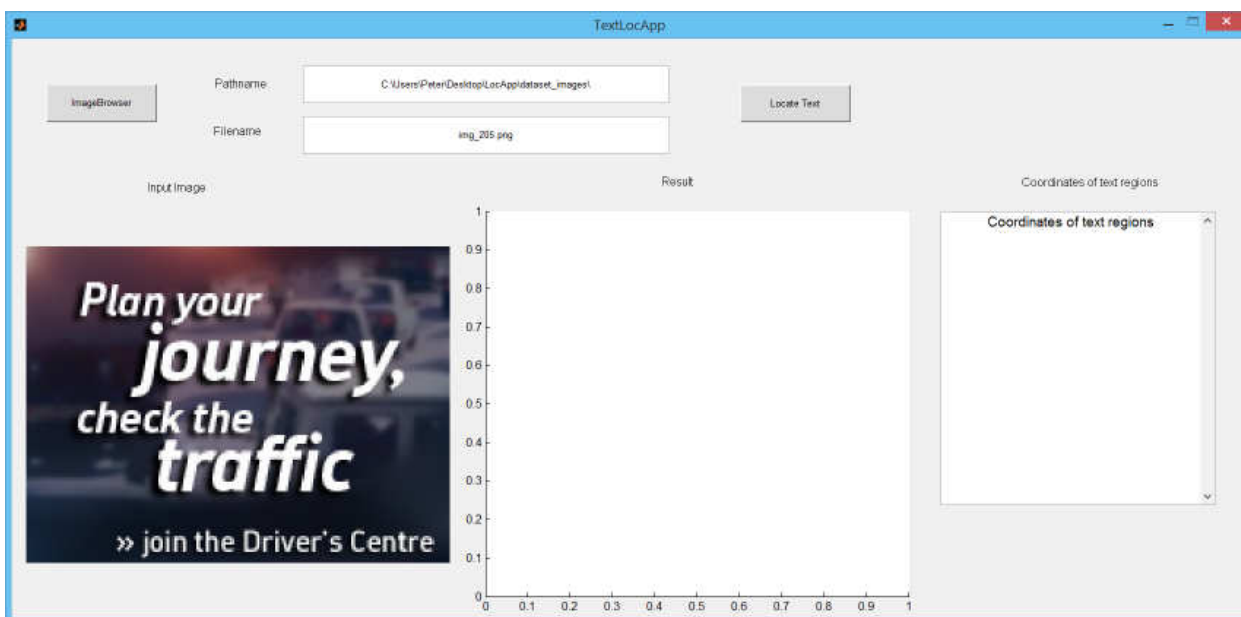


Рис. 4.6. Главное окно программы локализации с загруженным рисунком.

3. Для выполнения обнаружения и локализации ТО на изображении необходимо нажать кнопку “LocateText”. Полученный результат – координаты найденных ТО и изображение, которое содержит только найденные ТО, – отображается в главном окне программы (рис. 4.7).

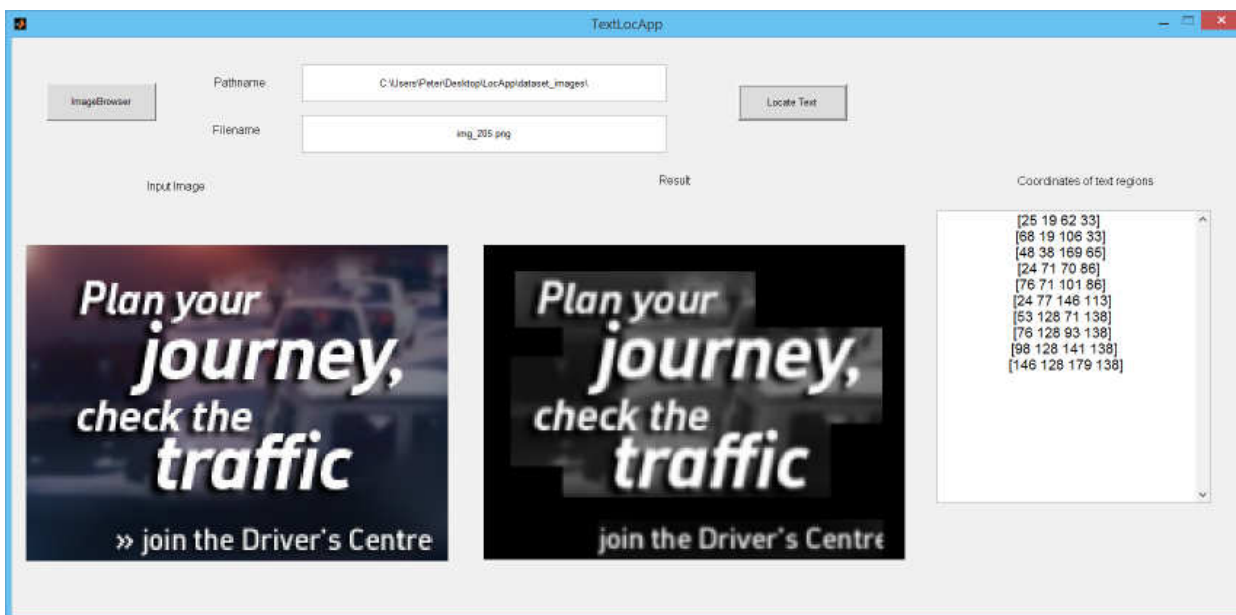


Рис. 4.7. Результат локализации.

Ниже рассмотрено более подробно функционирование программы ИТ локализации в соответствии с табл. 4.1.

В качестве исходного использовалось цветное RGB - изображение формата .png и размером $141 \times 189 \times 3$ пикселя (рис.4.8, а).



Рис. 4.8. Результат функционирования программы ИТ локализации:
а – исходное изображение; б – ТО-кандидаты; в – результат классификации найденных ТО

В соответствии с п. 2.1 (табл. 4.1) на основе применения корреляционно-экстремального метода обнаружения ТО найдены ТО-кандидаты (рис. 4.8, б). После применения усовершенствованного метода локализации к ТО-кандидатам осуществлена верификация ТО (см. п.п. 2.3, 2.4 табл. 4.2), локализованы ТО (рис. 4.8, в) и определены их координаты (табл. 4.3).

Таблица 4.3 Оценка качества локализации ТО разработанной программой

Надпись	Эталонные координаты ТО	Координаты ТО, полученные программой	Критерий Прэтта	Модифицированный критерий Прэтта
"plan"	24, 16, 61, 30	25, 19, 62, 33	0.587	0.673
"your"	65, 21, 104, 34	68, 19, 106, 33	0.318	0.906
"journey,"	47, 34, 167, 66	43, 38, 169, 65	0.266	0.554
"check"	23, 69, 69, 82	24, 71, 70, 86	0.219	0.615
"the"	75, 69, 100, 82	76, 71, 101, 86	0.266	0.661
"traffic"	59, 83, 145, 108	24, 77, 146, 113	0.065	0.110
"join"	51, 126, 70, 138	53, 128, 71, 138	0.488	0.903
"the"	75, 126, 92, 135	76, 128, 93, 138	0.292	0.684
"Driver's"	97, 126, 140, 135	98, 128, 141, 138	0.222	0.618
"Centre"	145, 126, 181, 135	146, 128, 179, 138	0.204	0.615

На основании результатов, представленных в табл. 4.3 проведена оценка качества локализации ТО с использованием разработанной ИТ. Для такой оценки использованы эталонные координаты ТО, взятые из базы данных конкурса алгоритмов искусственного интеллекта ICDAR 2011 Robust Reading Competition [108], и координаты этих же ТО, локализованных с помощью предложенной ИТ. Координаты ТО задаются в виде координат верхнего левого и нижнего правого угла ограничивающего ТО прямоугольника в соответствии с формулой (1.2).

В качестве примера на рис. 4.9 представлен результат локализации надписи "plan" (см. рис. 4.8 и табл. 4.3) и вид границ в соответствии с

эталонными координатами (надписи на рисунке «эталон», «программа» указывают на соответствующую границу, по осям отложены координаты в пикселях).

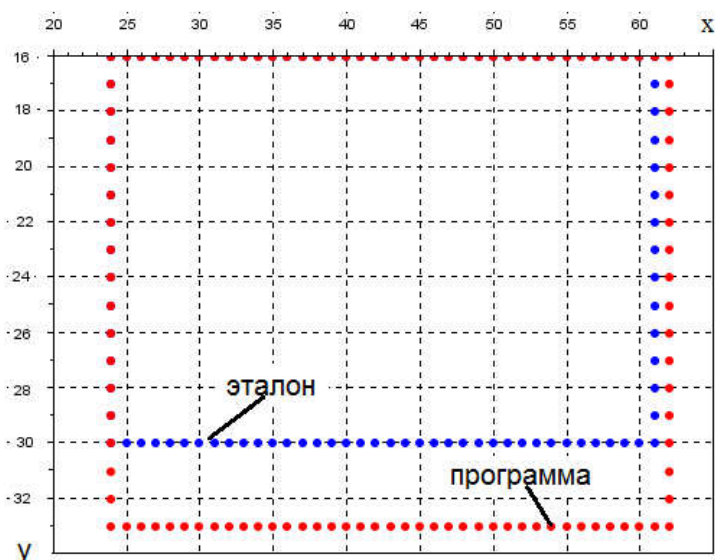


Рис. 4.9. Результаты локализации ТО "plan"

Оценка качества локализации проводилась на основании расчета значений критерия Прэтта (1.5). Значения критерия Прэтта, рассчитанные по формуле (1.5), представлены в столбце «Критерий Прэтта» (см. табл. 4.3). Следует отметить, что для обнаружения локализации ТО не нужна точность в один пиксель, так как локализованная ТО в дальнейшем должна сегментироваться на области изображений отдельных символов для распознавания последних. Несмотря на низкие значения критерия Прэтта погрешность локализации ТО лежит в пределах ± 4 пикселя, что не превышает половины изображения символа на малых масштабах.

Поэтому для оценки качества локализации предлагается модифицировать критерий Прэтта таким образом, чтобы расстояние между точками контура учитывалось только при условии, если оно превышает некоторый порог в

пикселях.

Модифицированный критерий Прэтта записывается в виде

$$F_{loc}^{mod} = \frac{1}{\max(N_t, N_{et})} \sum_{i=1}^{N_t} \frac{1}{1 + d_{mod}^2(i)},$$

где N_{et} – количество пикселей в множестве I_{et} ;

N_t – количество пикселей в множестве I_t ;

$d_{mod}^2(i) = 1(d^2(i) \geq tr) \cdot d^2(i)$ – модифицированное расстояние в пикселях;

$1(x)$ – единичная функция Хевисайда;

tr – заданный порог в пикселях;

$d^2(i) = \min(I_t(i) - I_{et}(j))^2$, $j = \overline{1, N_t}$ – квадрат расстояния от i -го пикселя множества I_t к ближайшему пикселю множества I_{et} .

В частности, значения такого модифицированного критерия для порога $tr = 2$ пикселя представлены выше (см. табл. 4.3, столбец «Модифицированный критерий Прэтта»). Такое введение критерия соответствует восприятию человека, для которого разница в 2 пикселя практически не заметна.

4.3. Тестирование работы свёрточной нейронной сети при локализации текстовых областей

С помощью разработанного программного комплекса проводилась оценка качества локализации ТО на изображениях со сложным фоном. Для этого использованы мера точности, мера отзыва, ошибка первого рода и ошибка второго рода (см. раздел 1, п.1.7).

Контрольная выборка для выполнения тестирования сформирована из 500

изображений сложных графических сцен, полученных из новостных или рекламных телевизионных роликов, фильмов, а также изображений вагонов, автомобилей и т.д. с помощью инструмента поиска изображения Google и базы данных конкурса алгоритмов искусственного интеллекта ICDAR 2011 Robust Reading Competition [108].

Результаты оценки качества обнаружения и локализации ТО на изображениях с использованием СНС представлены в табл. 4.4.

Таблица 4.4 – Результаты оценки качества обнаружения текстовых областей

Условия формирования обучающей выборки		Pr, %	Rc, %	Ошибка I рода, %	Ошибка II рода, %
без использования вероятностной модели представления символа текста		77,7	80,0	22,3	20,0
с использованием вероятностной модели представления символа	полутонные изображения	87,7	86,6	12,1	12,9
	цветные изображения	87,3	87,2	12,4	12,7

Примеры локализации текстовых областей на реальных изображениях, полученные с использованием разработанной информационной технологии, приведены на рис. 4.10, рис. 4.11.

Для сравнения качества локализации ТО было обработано 396 изображений со сложным фоном из телевизионных роликов, фильмов, рекламных баннеров из база данных конкурса алгоритмов искусственного интеллекта ICDAR 2011 [108] с помощью предложенной ИТ, программы FineReader11 и онлайн-программы FreeOCR (onlineOCR.net). Для оценки качества ИТ локализации ТО определены следующие показатели: Precision (Pr, %), Recall (Rc, %), ошибки I-ого и II-ого рода (%). Сравнительная оценка

эффективности работы ИТ локализации текстовых областей представлена на рис. 4.12.

Как видно из рис. 4.12 предложенная ИТ имеет результат, сравнимый с программой FineReader11 по показателю полноты Recall (процент правильно обнаруженных ТО по отношению к их общему количеству), однако по показателю Precision (процент правильно обнаруженных ТО по отношению к общему количеству обнаруженных областей, признанных как текстовые) в два раза превосходит программу FineReader11. Иными словами, программа FineReader11 ошибочно признавала как текстовые большее количество областей, по сравнению с предложенной ИТ (ошибка II-го рода – ложная тревога выше).

Программа FreeOCR (onlineOCR.net) по всем параметрам уступила предложенной ИТ. Для сравнения по отношению к FineReader11 она обеспечивает лучший показатель Precision, но существенно уступает по показателю Recall (т.е. имеет место много пропусков ТО – большое количество ошибок Recall I-го рода).



а)



б)

Рис.4.10. Пример локализации ТО на цветном изображении.

а – исходное изображение; б – результат локализации текстовой области

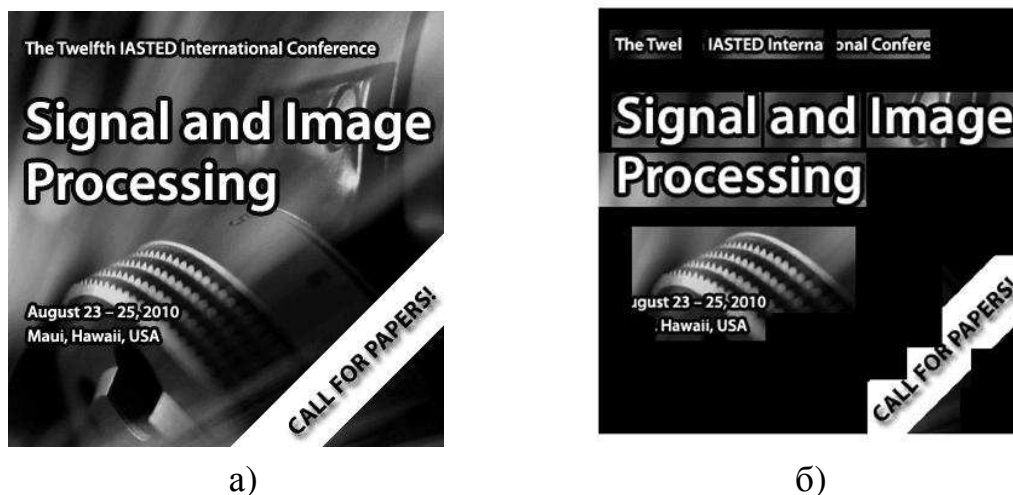


Рис.4.11 Пример локализации ТО на полутоновом изображении с применением предложенной ИТ:

а – исходное изображение; б – результат локализации текстовой области

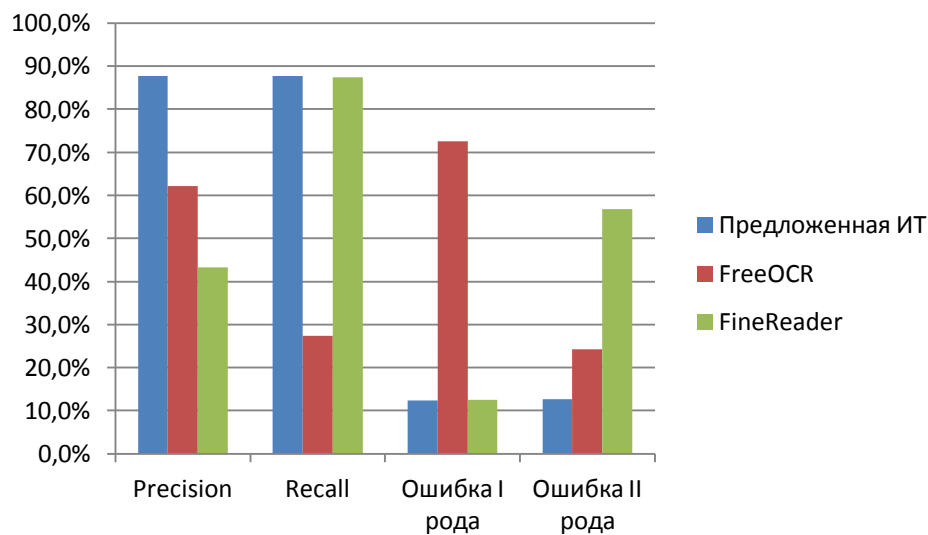
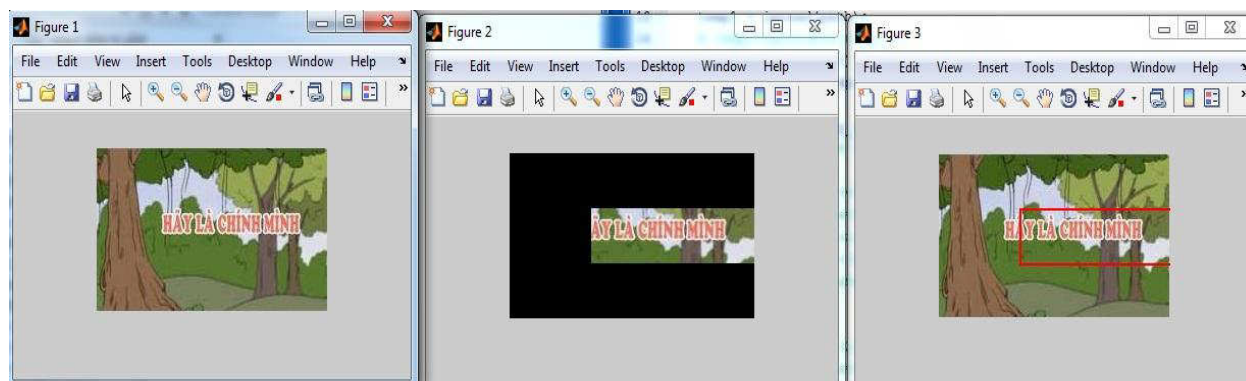


Рис. 4.12. Сравнительная оценка достоверности локализации текстовых областей

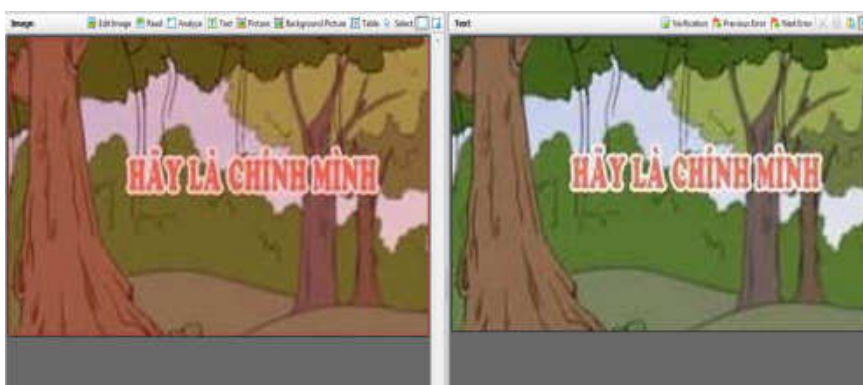
Качество локализации 396 изображений со сложным фоном телевизионных роликов, фильмов, рекламных баннеров с помощью предложенной информационной технологии составило 87.7 % (Precision), 86,6% (Recall), что превосходит результат, полученный с помощью программ

FineReader11 и FreeOCR.

Иллюстрация локализации ТО на сложном фоне с помощью предложенной ИТ и программы FineReader11 представлена на рис. 4.13.



а



б

Рис.4.13 Иллюстрация локализации ТО на сложном фоне с помощью:
а – предложенной технологии; б – программой FineReader11 (ТО не обнаружена).

Разработанные модели, методы и ИТ локализации ТО на сложном фоне были использованы для моделирования и разработки программно-алгоритмического обеспечения при автоматизации учета накопленной графической и видеоинформации в научно-производственном объединении «Дискрет» (г. Одесса), а именно модификации системы автоматизации проверки качества упаковки продукции и правильности распределения ее при формировании партий товара.

Проведение автоматизированной проверки качества упаковки продукции и правильности распределения ее при формировании партий товара позволило повысить эффективность функционирования склада продукции на предприятии, что подтверждается соответствующим актом внедрения.

4.4. Выводы по разделу 4.

1. Разработана информационная технология локализации ТО на основе предложенных методов обнаружения и локализации текстовых областей на изображении. Качество локализации на изображениях со сложным фоном телевизионных роликов, фильмов, рекламных баннеров с помощью предложенной информационной технологии составила 87,7% (точность), 86,6% (полнота), что превосходит результат, полученный с помощью программ FineReader11 и FreeOCR.

Оценка качества обнаружения ТО на контрольной выборке из 500 изображений со сложным фоном с применением разработанного корреляционно-экстремального метода обнаружения текстовых областей, использующего многомасштабное представление изображения на основе двумерного дискретного вейвлет-преобразования и сверточную нейронную сеть, и обучающим набором на основе созданных текстовых областей составила 87,7%, что на 10% выше качества локализации без использования обучающих наборов на основе разработанного метода формирования ТО (77,7%);

Качество локализации 396 изображений со сложным фоном телевизионных роликов, фильмов, рекламных баннеров с помощью предложенной информационной технологии составило 87,7 % (Precision), 86,6% (Recall), что превосходит результат, полученный с помощью программ FineReader11 и FreeOCR.

2. Информационная технология локализации текстовых областей для автоматической переработки графической информации внедрена при моделировании и разработке программно-алгоритмического обеспечения для автоматизации учета накопленной графической и видеоинформации в научно-производственном объединении «Дискрет» (г. Одесса). Аппробация подтвердила эффективность полученных решений, что подтверждается соответствующим актом внедрения.

3. Полученные в работе научные результаты и программные средства и внедрены в учебный процесс кафедры информационных систем Одесского национального политехнического университета при изучении дисциплин «Методы и средства компьютерных информационных технологий», «Методы и средства искусственного интеллекта», «Вычислительный интеллект» и при выполнении курсового проектирования.

ВЫВОДЫ

В диссертационной работе решена актуальная научно-практическая задача разработки информационной технологии локализации текстовых областей на изображениях со сложным фоном с целью повышения качества и оперативности локализации текстовых областей. При этом получены следующие основные результаты:

1. Анализ существующих информационных технологий, моделей и методов локализации текстовых областей на изображениях со сложным фоном показал, что качество локализации ТО на картинке существенно снижается в зависимости от сложности и неоднородности фона и влияния различных факторов. Предложено локализацию текстовых областей рассматривать как классификацию некоторой области изображения на два класса: «текстовая область» или «не текстовая область». Избран корреляционно-экстремальный подход как базовый для выявления текстовых областей на изображении благодаря его лучшей достоверности обнаружения ТО.

2. Усовершенствована вероятностная модель представления символа текста и предложен метод формирования изображений текстовых областей для создания представительной обучающей выборки для обучения классификаторов. Это позволило автоматизировать процесс создания представительной выборки, устранило субъективное влияние на обучение различных классификаторов для выявления текстовых областей на изображениях. Показано, что использование предложенного метода формирования текстовых областей на изображениях на основе использования вероятностной модели представления символа текста при формировании обучающей выборки за счет автоматизации процесса создания обучающей выборки существенно уменьшило время его формирования (в среднем на два

порядка), что повысило оперативность обнаружения и локализации текстовых областей на изображениях.

3. Предложено корреляционно-экстремальный метод обнаружения текстовых областей на изображениях на основе двумерного дискретного вейвлет-преобразования. Это позволило при сохранении высокой достоверности обнаружения ТО повысить оперативность локализации и уменьшить зависимость результатов обнаружения ТО от размеров и типов шрифтов текста.

4. Для реализации предложенного корреляционно-экстремального метода выявления текстовых областей разработана структура сверточной нейронной сети с многомасштабным представлением изображения на основе двумерного дискретного вейвлет-преобразования, что позволило уменьшить зависимость результатов локализации текстовой области от размеров и типов шрифтов текста. Обеспечено качество обнаружения текстовых областей контрольной выборки (500 изображений на сложном фоне) на 10% выше качества обнаружения ТО сетью без использования учебных наборов на основе созданных текстовых областей.

5. Усовершенствован метод локализации текстовых областей с использованием сверточной нейронной сети, дискретного вейвлет-преобразования и морфологических операций, что позволило обеспечить локализацию текстовых областей для различных размеров шрифтов. Использование предложенного метода локализации позволили повысить точность локализации на 10,5% по сравнению с результатами после корреляционно-экстремального метода выявления ТО.

6. Разработана информационная технология локализации ТО на основе предложенных методов обнаружения и локализации текстовых областей на изображении. Качество локализации на изображениях со сложным фоном телевизионных роликов, фильмов, рекламных баннеров с помощью

предложенной информационной технологии составило 87,7% (точность), 86,6% (полнота), что превосходит результат с программами FineReader11 и FreeOCR.

7. Полученные в работе научные результаты и программные средства использованы для моделирования и разработки программно-алгоритмического обеспечения в научно-производственном объединении «Дискрет» (г. Одесса) и внедрены в учебный процесс кафедры информационных систем Одесского национального политехнического университета.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Нгуен, Т. Т. К. Обнаружение и распознавание текстов на изображениях сложных графических сцен с помощью сверточной нейронной сети / Т. Т. К. Нгуен // *Електротехнічні та комп'ютерні системи*. – 2014. – № 13(89). – С. 125 – 130.
2. Антощук, С. Г. Локализация квазипериодических последовательностей на изображениях / С. Г. Антощук, А. А. Николенко, Тьен Т. К. Нгуен // *Компьютерные науки и технологии*. – 2009. – № 2. – С. 35 – 40.
3. Николенко, А. А. Обнаружение текстовых областей в видеопоследовательностях / А. А. Николенко, Т. Т. К. Нгуен // *Искусственный интеллект*. – 2012. – № 4. – С. 227 – 234.
4. Николенко, А. А. Выбор масштаба вейвлет-преобразования при локализации квазипериодических последовательностей / А. А. Николенко, Е. В. Ткаченко, Т. Т. К. Нгуен // *Современные информационные и электронные технологии: труды XI Междунар. научно-практ. конф.: 24 – 28 мая 2010 г.* – Одесса, 2010. – Т. 1. – С. 54.
5. Николенко, А. А. Обнаружение текстовых фрагментов на видеоизображениях / А. А. Николенко, Тьен Т. К. Нгуен // *Современные информационные и электронные технологии (СИЭТ – 2012) : труды XIII Междунар. научно-практ. конф.: 4 – 8 июня 2012 г.* – Одесса, 2012. – С. 114.
6. Ніколенко, А. Метод виявлення текстових областей в відеопослідовності з використанням вейвлет-перетворення / [А. Ніколенко, О. Бабілонга, Тьен Т. К. Нгуен, В. Зайковський // *Оброблення сигналів і зображень та розпізнавання образів (УкрОБРАЗ'2012) : XI Всеукр. міжнарод. конф. : зб. наук. праць.* – Київ : МННЦІТС, 2012. – С. 99 – 102.

7. Николенко, А. А. Обнаружение текстовых областей в видеопоследовательностях / А. А. Николенко, Т. Т. К. Нгуен // Искусственный интеллект. Интеллектуальные системы («ИИ-2012»): материалы Междунар. научно-технической конференции: 1–5 октября 2012 г. – АР Крым, 2012. – С.115–118.

8. Николенко, А. А. Обнаружение текстовых областей и выделение символов на изображениях с неоднородным фоном /А. А. Николенко, А. В. Котляр, Т. Т. К. Нгуен // Праці Одеського політехнічного університету. – 2013. – Вип. 1(40).– С.55–60.

9. Николенко, А. А. Обнаружение и выделение символов текста на изображениях с неоднородным фоном / А. А. Николенко, А. В. Котляр, Т. Т. К. Нгуен // Современные информационные и электронные технологии: труды XIV Междун. научно-практ. конф.: 27 – 31 мая 2013 г. – Одесса, 2013. – С. 55–57.

10. Kotlyar, A. V. Text detection and characters extraction in images with non-uniform background / A. V. Kotlyar, Tien T. K. Nguyen // Сучасні інформаційні технології (МІТ – 2013) : III Міжнарод. наук. конф. студентів та молодих вчених ; тези доп., 25 – 26 квітня 2013 р., Одесса : ОНПУ, 2013. – С. 108 – 109.

11. Николенко, А. А. Локализация текстовых областей на изображениях с использованием сверточной нейронной сети / А. А. Николенко, О. Ю. Бабилунга, Т. Т. К. Нгуен // Вісник Національного технічного університету «Харківський політехнічний інститут». Серія : Інформатика та моделювання : зб. наук. праць. – Харків : НТУ «ХПИ». – 2013. – № 19 (992). – С. 121 – 127.

12. Николенко, А. А. Локализация текстовых областей на изображениях с использованием сверточной нейронной сети / А. А. Николенко, О. Ю. Бабилунга, Тьен Т. К. Нгуен // Проблемы информатики и моделирования : XIII Междун. научн.-техн. конф. : 23 – 29 сентября 2013 г, тезисы докл. – Харьков : НТУ «ХПИ». – 2013. – С. 52.

13. Николенко, А. А. Распознавание текстов на изображениях сложных графических сцен / А. А. Николенко, Тьен Т. К. Нгуен, П. В. Гордиенко // Современные информационные и электронные технологии (СИЭТ – 2014) : труды XV Междунар. научно-практ. конф. : 26 – 30 мая 2014 г. – Одесса, 2014. – С. 55 – 57.

14. Антощук, С. Г. Метод локализации текста на изображении с обучением на основе вероятностной модели символов / С. Г. Антощук, А. А. Николенко, Тьен Т. К. Нгуен // Вимірювальна та обчислювальна техніка в технологічних процесах. – Хмельницький, 2014. – ХНУ. – № 3. – С. 79 – 84.

15. Антощук, С. Г. Информационная технология локализации текстовых областей на сложном фоне / С. Г. Антощук, А. А. Николенко, Тьен Т. К. Нгуен // Электротехнические и компьютерные системы . – 2014. – № 16 (92). – С. 114 – 121.

16. Андрианов, А. И. Локализация текста на изображениях сложных графических сцен [Электронный ресурс] / А. И. Андрианов // Современные проблемы науки и образования. – 2013. – № 3. – Режим доступа : www.science-education.ru/109-9311 .

17. Coates, A. Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning Text / A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. Wu, A. Ng. // The 11th International Conference on Document Analysis and Recognition (ICDAR). – 2011. – P. 440 – 445.

18. Epshtein, B. Detecting Text in Natural Scenes with Stroke Width Transform Text. / B. Epshtein, E. Ofek, Y. Wexler // 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, 2010. – Vol. 5.

19. Kunishige, Y. Scenery Character Detection with Environmental Context Text. / Y. Kunishige, F. Yaokai, S. Uchida // The 11th International Conference on Document Analysis and Recognition (ICDAR). – 2011. – P. 1049 – 1053.

20. Uchida, S. A Keypoint-Based Approach Toward Scenery Character

Detection Text. / S. Uchida, Y. Shigeyoshi, Y. Kunishige, F. Yaokai // The 11th International Conference on Document Analysis and Recognition (ICDAR). – 2011. – P. 819 – 823.

21. Du, Y. Dot Text Detection Based on FAST Points Text. / Y. Du, H. Ai, S. Lao // The 11th International Conference on Document Analysis and Recognition (ICDAR). – 2011. – P. 435 – 439.

22. Jung, C. Accurate text localization in images based on SVM output scores Text / C. Jung, Q. F. Liu, J. Kim // Image and Vision Computing. – 2009. – Vol. 27. – P. 1295 – 1301.

23. Системы распознавания текста: [Электронный ресурс]. – Режим доступа : <http://www.softmart.ru/ru/sortiment/f23.htm> .

24. Системы распознавания текстовой информации: [Электронный ресурс]. – Режим доступа : <http://www.internet-school.ru/Enc.ashx?item=3754> .

25. Шамис, А. Л. Принципы интеллектуализации машинного распознавания изображений и их реализация в системах оптического чтения текстов – ABBYY FineReader и FormReader / А. Л. Шамис // Новости Искусственного интеллекта – 2002. – Вып. №1. – С.521 – 528.

26. Модуль распознавания номеров железнодорожных вагонов «Транзит-Инспектор» [Электронный ресурс]. – Режим доступа : <http://www.iss.ru/company/publicity/news/1317> .

27. Система «Авто-Инспектор»: [Электронный ресурс]. – Режим доступа : http://www.cam-bridge.ru/oborud/auto_inspector.html .

28. Система «Автоинтеллект»: [Электронный ресурс]. – Режим доступа : <http://www.efavideo.ru/avtoint.htm> .

29. Система «Бастион – Состав» предназначена для распознавания номеров и учета перемещений железнодорожных вагонов: [Электронный ресурс]. – Режим доступа : http://www.elsystems.ru/works/bastion_sostav .

30. Система «Карго-Инспектор-1» предназначена для распознавания номеров, нанесенных на крыше контейнеров: [Электронный ресурс]. – Режим доступа : <http://www.iss.ru/products/intelligent/cargo-inspector> .

31. Система «Синергет: учет железнодорожных вагонов»: [Электронный ресурс]. – Режим доступа : <http://www.videolocator.ru/rus/decisions/prom/sinerget-vagon> .

32. Система видеонаблюдения «КОДОС-АВТО» Распознавание номеров вагонов: [Электронный ресурс]. – Режим доступа : <http://www.intteks.com.ua/kavto.html> .

33. Система распознавания автомобильных номеров Auto-TRASSIR: [Электронный ресурс]. – Режим доступа : <http://www.centres.ru/id183/price/price166.htm> .

34. Система распознавания автомобильных номеров ДИГНУМ Авто : [Электронный ресурс]. – Режим доступа : <http://www.auto.dignum.ru> .

35. Система распознавания автомобильных номеров ПОТОК : [Электронный ресурс]. – Режим доступа : <http://www.rossi-potok.ru>

36. Система распознавания контейнеров Интегра-С : [Электронный ресурс]. – Режим доступа : <http://integra-s.by.ru/container.htm> .

37. Система распознавания маркировки контейнеров SeeContainer : [Электронный ресурс]. – Режим доступа : <http://www.ofeksystems.ru/products/recognition-systems> .

38. Система распознавания номеров железнодорожных вагонов и цистерн RailwayDisp : [Электронный ресурс]. – Режим доступа : http://www.intlab.ru/page.php?ipage=railwaydisp_info .

39. Система распознавания номеров контейнеров : [Электронный ресурс]. – Режим доступа : http://www.artvision.mallanom/ru/inno_vision_persp.php .

40. Система считывания номеров железнодорожных вагонов и цистерн : [Электронный ресурс]. – Режим доступа : <http://www.iitvision.ru/wagon.htm> .
41. Системы распознавания текста : [Электронный ресурс]. – Режим доступа : <http://www.softmart.ru/ru/sortiment/f23.htm> .
42. Системы распознавания текстовой информации : [Электронный ресурс]. – Режим доступа : <http://www.internet-school.ru/Enc.ashx?item=3754> .
43. Datong, C. Text detection and recognition in images and video sequences. / C. Datong. // Rapport de recherche de l'IDIAP 03-44. – 2003.– 129 p. [Электронный ресурс]. – Режим доступа : <http://publications.idiap.ch/downloads/reports/2003/rr03-44.pdf>
44. Крылов, В. Н. Повышение быстродействия определения границ сегмента упорядоченной текстуры на изображении с однородным фоном с помощью одноканального обнаружения пачки импульсов / В. Н. Крылов, М. В. Полякова // Оптико-электрон. інформ.-енерг. технології. – 2008. – №1. – С. 73 – 80.
45. Полякова, М. В. Определение границ сегмента упорядоченной текстуры на изображении с однородным фоном с помощью многоканального обнаружения пачки импульсов / М. В. Полякова // Автоматика. Автоматизация. Электротехнические комплексы и системы: межвузовский журнал. – Херсон: ХНТУ, 2008. – № 1 (21). – С. 72– 81.
46. Антощук, С. Г. Адаптивна локалізація символічних написів на зображеннях методом вейвлет-аналізу / С. Г. Антощук, А. О. Ніколенко, О. Ю. Бабілуंगा, О. В. Ткаченко // Вісник Житомирського технол. ун-ту. – 2008. – № 4 (47). – С. 124 – 130.
47. Антощук, С. Г. Моделирование интерпретации изображений в области вейвлет преобразования / С. Г. Антощук, А. А. Николенко // Електромашинобудування та електрообладнання. – 2010. – Вип. 75. – С. 121 – 125.

48. Антощук, С. Г. Использование вейвлет-преобразования при локализации последовательностей символов / С. Г. Антощук, А. А. Николенко, Е. В. Ткаченко, О. Ю. Бабилунга // Искусственный интеллект. – 2009. – № 4. – С. 23 – 29.

49. Martin, A. Image Processing Techniques for Machine Vision. / A. Martin, S. Tosunoglu // 2000. – [Электронный ресурс] : – Режим доступа :

http://www.eng.fiu.edu/mme/robotics/elib/am_st_fiu_ppr_2000.pdf

50. Распознавание символов по скелетному изображению на основе нейронной сети / [А. Д. Варламов, М. Ю. Арефьева, Ю. А. Севитова, Ю. С. Степанова] // Сигнейро - технологии обработки и анализа изображений нейронными сетями – [Электронный ресурс] – Режим доступа:

http://www.signneuro.ru/item2_2.php

51. Nikolaou, N. Segmentation of historical machine-printed documents using Adaptive Run Length Smoothing and skeleton segmentation paths / N. Nikolaou, M. Makridi, B. Gatos, N. Stamatopoulos, N. Papamarkos // Image and Vision Computing. – 2010. – Vol. 28. – P. 590 – 604.

52. Sengupta, S. and Hyun S. Yang. Morphological Shape Representation and Recognition of Binary Images / S. Sengupta, and Hyun S. Yang // Proceedings of SPIE. – 1987. – Vol. – 0848. – P. 51-56.

53. Петровцева М.А., Рейер И.А. Язык гранично-скелетного представления бинарных изображений // Труды 13-й международной конференции по компьютерной графике и машинному зрению "Графикон-2003", Москва, 2003, С. 228-234

54. Фраленко, В. П. Локализация текстовых фрагментов на смешанном фоне: краткий научный обзор / В. П. Фраленко // Программные системы : теория и приложения : электрон. научн. журн. – 2014. – Т. 5. – № 2 (20). – С. 33 – 45. – [Электронный ресурс]. – Режим доступа : http://psta.psisras.ru/read/psta2014_2_33-45.pdf .

55. Погодин, С. В. Выделение и анализ скелетов объектов на цветных снимках / С. В. Погодин // Программные продукты и системы. – 2009. – Т. 2. – С. 42 – 45.
56. Шапиро, Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман. – М. : БИНОМ. Лаб. знаний. – 2006. – 752 с.
57. Абрамов, Н. С. Распознавание на основе инвариантных моментов / Н. С. Абрамов, В. М. Хачумов // Вестник РУДН. Серия : Математика. Информатика. Физика. – 2014. – Т. 2. – С. 142 – 149.
58. Виноградов, А. Н. Выделение и распознавание локальных объектов на аэрокосмических снимках / А. Н. Виноградов, Ф. В. Калугин, М. Д. Недев // Авиакосмическое приборостроение. – 2007. – Т. 9. – С. 39 – 45.
59. Завалишин, Н. В. Модели зрительного восприятия и алгоритмы анализа изображений / Н. В. Завалишин, И. Б. Мучник // Наука. – 1974. – 344 с.
60. Талалаев, А. А. Выделение и кластеризация текстовых и графических элементов на полутоновых снимках / А. А. Талалаев, И. П. Тищенко, М. В. Хачумов // Искусственный интеллект и принятие решений. – 2008. – Т. 3. – С. 72 – 84.
61. Yang, M. Recent Advances in Face Detection / M. Yang // Tutorial of The IEEE International Conference on Pattern Recognition. – Cambridge (United Kingdom), 2004. – 93 p.
62. Haralick, R. M. Textural features for image classification / R. M. Haralick, K. Shanmugam, I. Dinstein // IEEE Trans. on Systems, Man and Cybernetics. – 1973. – Vol. 3. – P. 610 – 621.
63. Рубан, И. В. Метод обработки цифрового изображения для локализации информативных областей / И. В. Рубан, О.В. Шитова, Д. П. Пашков // Моделювання та інформаційні технології: Зб. наук. пр. – К.: ІПМЕ ім. Г. Є. Пухова НАН України, 2010. – Вип. 55. – С. 225-229.

64. Anthimopoulos, M A two-stage scheme for text detection in video images/ M. Anthimopoulos, B. Gatos, I. Pratikakis // Image and Vision Computing, 2010. – № 28. – P. 1413 – 1426.
65. Chen, H. Robust text detection in natural images with edge-enhanced Maximally Stable Extremal Regions / H. Chen, S. Chen, S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk and B. Girod // Image Processing (ICIP'2011) : 18th IEEE International Conference, 11 – 14 Sept. 2011. – P. 2609 – 2612.
66. Chen, D. Text detection, recognition in images and video frames / D. Chen, J. M. Odobez, H. Bourlard // Pattern Recognition. – Vol. 37(3). – P. 595 – 608. – 2004.
67. Ciresan, D. C. High-performance neural networks for visual object classification / D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber // Istituto Dalle Molle di Studisull'Intelligenza Artificiale (IDSIA), Technical Report No. IDSIA-01-11, January 2011.
68. Chucai, Y. Localizing Text in Scene Images by Boundary Clustering, Stroke Segmentation, and String Fragment Classification / Y Chucai, T. Yingli // IEEE Transactions on Image Processing. –2012.– Vol. 21.– issue 9. – P. 4256-4268
69. Delakis, M. Text detection with convolutional neural networks / M. Delakis, Cr. Garcia // International Conference on Computer Vision Theory and Applications. – 2008. – P. 290 – 294.
70. Jain, V. Natural image denoising with convolutional networks / V. Jain and H. S. Seung // Advances in Neural Information Processing Systems (Proceedings of NIPS '08). – 2009.– P. 769 – 776.
71. Canny, J. A Computational Approach to Edge Detection / J. Canny // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1986. – Vol. 8, No. 6. – P. 679-698.

72. Augusteijn, M. Identification of Human Faces through Texture-Based Feature Recognition and Neural Network Technology / M. Augusteijn, T. Skujca // Proceedings of The IEEE Conference on Neural Networks. – 1993. – P. 392-398.
73. Блаттер, К. Вейвлет-анализ. Основы теории / К. Блаттер. — М. : Техносфера, 2006. — 523 с.
74. Добеши, И. Десять лекций по вейвлетам / И. Добеши. — М. : Регулярная и хаотическая динамика. – 2001. – 464 с.
75. Короновский, А. А. Непрерывный вейвлетный анализ и его приложения / А. А. Короновский, А. Е. Храмов – М. : Физматлит, 2003. – 176 с.
76. Малла, С. Вейвлеты в обработке сигналов / С. Малла. – М. : Мир. – 2005. – 671 с.
77. Смоленцев, Н. К. Основы теории вейвлетов / Н. К. Смоленцев – М. : Пресс, 2008. — 448 с.
78. Столниц, Э. Вейвлеты в компьютерной графике / Э. Столниц, Т. Де Роуз, Д. Салезин – Ижевск : «Реглярная и хаотическая динамика», 2002. – 272 с.
79. Штарк, Г. Г. Применение вейвлетов для ЦОС / Г. Г. Штарк — М. : Техносфера, 2007. — 192 с.
80. Гонсалес, Р. Обработка изображений в среде Matlab / Р. Гонсалес, Р. Вудс, С. Эддинс. – М. : Техносфера. – 2006. – 616 с.
81. Клих, Ю. А. Адаптивные базисные функции вейвлетного преобразования / Ю. А. Клих, С. Г. Антощук, А. А. Николенко // Труды Одес. политехн. ун-та. – 2004. – №2(22). – С. 121 – 125.
82. Модель формирования образов при распознавании полутоновых изображений / [С. Г. Антощук, А. А. Николенко, О. Ю. Бабилунга, Е. В. Ткаченко] // Радіоелектронні і комп'ютерні системи. –2008. – № 6 (33). – С. 283 – 287.
83. Полякова, М. В. Морфологический метод контурной сегментации изображений на основе репагулярного вейвлет-преобразования /

М. В. Полякова, В. Н. Крылов // Тр. Одес. политехн. ун-та. – 2006. – № 1(25). – С. 98 – 103.

84. Antoshchuk, S. The Hierarchical Objects Analysis on Images of Cutting Tool Wear Zones / S. Antoshchuk, A. Derevianchenko, E. Tkachenko // Сучасні проблеми радіоелектроніки, телекомунікацій, комп'ютерної інженерії : Між нар. конф., 28 лютого - 4 березня 2006 р.: зб. наук. праць. – Львів : «Львівська політехніка». – 2006. – С. 253 – 255.

85. Pratt, W.K. Digital image processing / Pratt W.K. – USA : John Wiley & Sons, 2001. – 750 p.

86. Konishi, S. A Statistical Approach to MultiScale Edge Detection / S. Konishi, A. L. Yuille, J. M. Coughlan // Proc. Workshop Generative-Model-Based Vision : GMBV'02, 2002. – P. 837 – 843.

87. Audithan, S. “Document Text Localization from Document Images Using Haar Discrete Wavelet Transform” / S. Audithan, R.M. Chandrasekaran // European Journal of Scientific Research ISSN 1450-216X. – Vol. 36 (2009).

88. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – М. : Вильямс. – 2006. – 1104 с.

89. Друки, А. А. Применение сверточных нейронных сетей для выделения и распознавания автомобильных номерных знаков на изображениях со сложным фоном / А. А. Друки // Известия Томского политехнического университета. – 2014. – Т. 324. – № 5. – С. 85–92.

90. Болотова, Ю. А. Распознавание символов на цветном фоне на основе иерархической временной модели с предобработкой фильтрами Габора / Ю. А. Болотова, В. Г. Спицын, А. К. Кермани // Электромагнитные волны и электронные системы. – 2012. – Т. 16. – № 1. – С. 14–19.

91. Bundzel, M. Object identification in dynamic images based on the memory prediction theory of brain function / M. Bundzel, S. Hashimoto // Journal of

Intelligent Learning Systems and Applications. – 2010. – Vol. 2. – No. 4. – P. 212 – 220.

92. Чанг, Б. Т. Т. Алгоритмическое и программное обеспечение для классификации цифровых изображений с помощью вейвлет-преобразования Хаара и нейронных сетей / Буй Тхи Тху Чанг, Фан Нгок Хоанг, Спицын В.Г. // Известия Томского политехнического университета. – 2011. – Т. 319. – № 5. – С. 103–106.

93. LeCun, Y. Gradient-based learning applied to document recognition / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // Proceedings of the IEEE. – 1998. – Vol. 86 (11). – P. 2278 – 2324.

94. Умяров, Н. Х. Параметрическая модель свёрточной нейронной сети / Н. Х. Умяров, О. И. Федяев // Информатика и компьютерные технологии : VI Междунар. научно-техн. конф. студентов, аспирантов и молодых научных работников. – Донецк : ДонНТУ. – 2010. – Т. 2. – С. 292.

95. Умяров, Н. Х. Логическое и физическое представление архитектуры свёрточной нейронной сети / Н. Х. Умяров, О. И. Федяев // II Всеукр. наук.-техн. конф. студентів, аспірантів та молодих вчених : зб. наук. праць у 3 т., 11 – 13 квітня 2011 р., Донецьк / Донец. націонал. техн. ун-т. – Донецьк : ДонНТУ. – 2011. – Т.3. – С. 81 – 85.

96. Умяров, Н. Х. Программная модель сверточной нейронной сети / Н. Х. Умяров, Г. Ю. Костецкая, О. И. Федяев // Информационные управляющие системы и компьютерный мониторинг : III Всеукр. наук.-техн. конф. студентів, аспірантів та молодих вчених : зб. наук. праць, 16 – 19 квітня 2012 р., Донецьк : ДонНТУ. – 2012. – С. 343 – 347.

97. Shesagiri, T. End-to-End Text Recognition with Convolutional Neural Networks / T. Shesagiri, N. Upadhayaya, G.VijayaKumari // International Journal of Latest Research in Science and Technology. –2012.– Vol. 1.–Issue 2. P.132-136.

98. Hansen, D.W. Eye typing using Markov and active appearance models / D. W. Hansen, J. P. Hansen, M. Nielsen // Workshop on applications of computer vision. – Orlando, Florida, USA, IEEE Computer Society. – 2002. – Vol. 12. – P. 132 – 136.
99. Lawrence, S. Face Recognition: a Convolutional Neural Network Approach / S. Lawrence, C. Giles, A. Tsoi, A. Back // IEEE Transactions on Neural Networks: (Special Issue on Neural Networks and Pattern Recognition). – 1997. – Vol. 8, Iss. 1. – P. 98-113.
100. Garcia, C. Convolution Face Finder: A Neural Architecture for Fast and Robust Face Detection / C. Garcia, M. Delakis // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2004. – Vol. 26, Iss. 11. – P. 1408-1423.
101. Delakis, M. Text detection with convolutional neural networks / M. Delakis Delakis and C. Garcia // VISAPP 2008 – International Conference on Computer Vision Theory and Applications. – P. 290-294. – [Электронный ресурс] .
Режим доступа : <http://iris.cnrs.fr/Documents/Liris-6079.pdf>
102. Палій, І. О. Методи виявлення обличчя людини на основі комбінованого каскаду класифікаторів / І. О. Палій // Комп'ютинг. – 2008. – Том 7, Вип. 1. – С. 114-125.
103. Simard, P.Y. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis / P.Y. Simard, D. Steinkraus, J. Platt // International Conference on Document Analysis and Recognition (ICDAR).– IEEE Computer Society. – Los Alamitos. – 2003. – P. 958-962.
104. Прохоров, В. Г. Использование свёрточных сетей для распознавания рукописных символов // Проблемы програмування.– 2008.– № 2-3.– С.669–674.
105. Rumelhart, D. E. Learning Internal Representations by Error Propagation / D. E. Rumelhart, G. E. Hinton, R. J. Williams // Parallel Distributed Processing.– Cambridge, MA, MIT Press. – 1986. –Vol. 1. – P. 318–362.

106. Дорогой, Я. Ю. Архитектура обобщенных сверточных нейронных сетей / Я. Ю. Дорогой // Вісник НТУУ «КПІ», «Інформатика, управління та обчислювальна техніка». – 2001.– №54. – С. 229–234.

107. Солдатова О. П. Применение сверточной нейронной сети для распознавания рукописных цифр / Солдатова О.П., Гаршин А.А. // Самарский государственный аэрокосмический университет имени академика С. П. Королева. – Компьютерная оптика.– 2010.– Т 34.– №2.– С. 252-259

108. База данных конкурса алгоритмов искусственного интеллекта ICDAR 2011 Robust Reading Competition. – [Электронный ресурс]. – Режим доступа : <http://www.cvc.uab.es/icdar2011competition/>

109. Вежневц, В. Оценка качества работы классификаторов / В. Вежневц // Компьютерная графика и мультимедиа. – № 4. – 2006. – С. 106.

110. Powers, D. M W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation / D. M W Powers //Technical Report SIE-07-001. – 2007.– 24 p. – [Электронный ресурс]. – Режим доступа :

http://www.flinders.edu.au/science_engineering/fms/School-CSEM/publications/tech_reps-research_artfcts/TRRA_2007.pdf .

ПРИЛОЖЕНИЯ

Приложение А.

Фрагменты программного кода

%Структуры свёрточной нейронной сети для локализации текстовых %областей на цветных изображениях

```

classdef ConvNN
    properties

% Layer C1

        WC101R;          WC110G;          WC119B;
        WC102R;          WC111G;          WC120B;
        WC103R;          WC112G;          WC121B;
        WC104R;          WC113G;          WC122B;
        WC105R;          WC114G;          WC123B;
        WC106R;          WC115G;          WC124B;
        WC107R;          WC116G;          WC125B;
        WC108R;          WC117G;          WC126B;
        WC109R;          WC118G;          WC127B;

        biasC101;        biasC110;        biasC119;
        biasC102;        biasC111;        biasC120;
        biasC103;        biasC112;        biasC121;
        biasC104;        biasC113;        biasC122;
        biasC105;        biasC114;        biasC123;
        biasC106;        biasC115;        biasC124;
        biasC107;        biasC116;        biasC125;
        biasC108;        biasC117;        biasC126;
        biasC109;        biasC118;        biasC127;

% Layer S1

        WS1C101;
        WS1C110;
        WS1C119;

        biasS101;
        biasS110;
        biasS119

% Layer C2

        WC201_01;        WC203_11;        WC205_22;
        WC201_10;        WC203_20;        WC206_05;
        WC201_19;        WC204_03;        WC206_14;
        WC202_01;        WC204_12;        WC206_23;
        WC202_10;        WC204_21;        WC207_06;
        WC202_19;        WC205_04;        WC207_15;
        WC203_02;        WC205_13;        WC207_24;

```

```

WC208_07;          WC209_08;          WC210_09;
WC208_16;          WC209_17;          WC210_18;
WC208_25;          WC209_26;          WC210_27;
biasC201           biasC205           biasC209
biasC202           biasC206           biasC210
biasC203           biasC207
biasC204           biasC208

% layer S2
WS2C201;
WS2C202;
WS2C203;
WS2C204;
WS2C205;
WS2C206;
WS2C207;
WS2C208;
WS2C209;
WS2C210;

% layer N1
WN101             biasN101
WN102             biasN102
WN103             biasN103
WN104             biasN104
WN105             biasN105
WN106             biasN106
WN107             biasN107
WN108             biasN108
WN109             biasN109
WN110             biasN110

% layerN2
WN2
biasN2

end

methods
function cnet=ConvNN()
% Layer C1

cnet.WC101R=rand(25,32,60);
cnet.WC102R=rand(25,14,28);
cnet.WC103R=rand(25,14,28);
cnet.WC104R=rand(25,14,28);
cnet.WC105R=rand(25,14,28);
cnet.WC106R=rand(25,5,12);
cnet.WC107R=rand(25,5,12);
cnet.WC108R=rand(25,5,12);
cnet.WC109R=rand(25,5,12);

cnet.WC110G=rand(25,32,60);
cnet.WC111G=rand(25,14,28);
cnet.WC112G=rand(25,14,28);
cnet.WC113G=rand(25,14,28);
cnet.WC114G=rand(25,14,28);
cnet.WC115G=rand(25,5,12);
cnet.WC116G=rand(25,5,12);
cnet.WC117G=rand(25,5,12);

```

```

cnet.WC118G=rand(25,5,12);

cnet.WC119B=rand(25,32,60);
cnet.WC120B=rand(25,14,28);
cnet.WC121B=rand(25,14,28);
cnet.WC122B=rand(25,14,28);
cnet.WC123B=rand(25,14,28);
cnet.WC124B=rand(25,5,12);
cnet.WC125B=rand(25,5,12);
cnet.WC126B=rand(25,5,12);
cnet.WC127B=rand(25,5,12);

cnet.biasC101=rand(32,60);
cnet.biasC102=rand(14,28);
cnet.biasC103=rand(14,28);
cnet.biasC104=rand(14,28);
cnet.biasC105=rand(14,28);
cnet.biasC106=rand(5,12);
cnet.biasC107=rand(5,12);
cnet.biasC108=rand(5,12);
cnet.biasC109=rand(5,12);

cnet.biasC110=rand(32,60);
cnet.biasC111=rand(14,28);
cnet.biasC112=rand(14,28);
cnet.biasC113=rand(14,28);
cnet.biasC114=rand(14,28);
cnet.biasC115=rand(5,12);
cnet.biasC116=rand(5,12);
cnet.biasC117=rand(5,12);
cnet.biasC118=rand(5,12);
cnet.biasC119=rand(32,60);
cnet.biasC120=rand(14,28);
cnet.biasC121=rand(14,28);
cnet.biasC122=rand(14,28);
cnet.biasC123=rand(14,28);
cnet.biasC124=rand(5,12);
cnet.biasC125=rand(5,12);
cnet.biasC126=rand(5,12);
cnet.biasC127=rand(5,12);

% Layer S1
cnet.WS1C101=0.25*ones(16,30);
cnet.WS1C110=0.25*ones(16,30);
cnet.WS1C119=0.25*ones(16,30);

cnet.biasS101=rand(16,30);
cnet.biasS110=rand(16,30);
cnet.biasS119=rand(16,30);

% Layer C2
cnet.WC201_01=rand(9,14,28);
cnet.WC201_10=rand(9,14,28);
cnet.WC201_19=rand(9,14,28);

cnet.WC202_01=rand(9,14,28);
cnet.WC202_10=rand(9,14,28);
cnet.WC202_19=rand(9,14,28);

cnet.WC203_02=rand(9,12,26);
cnet.WC203_11=rand(9,12,26);
cnet.WC203_20=rand(9,12,26);

cnet.WC204_03=rand(9,12,26);
cnet.WC204_12=rand(9,12,26);
cnet.WC204_21=rand(9,12,26);

cnet.WC205_04=rand(9,12,26);
cnet.WC205_13=rand(9,12,26);
cnet.WC205_22=rand(9,12,26);

cnet.WC206_05=rand(9,12,26);
cnet.WC206_14=rand(9,12,26);
cnet.WC206_23=rand(9,12,26);

cnet.WC207_06=rand(9,3,10);

cnet.WC207_15=rand(9,3,10);
cnet.WC207_24=rand(9,3,10);

cnet.WC208_07=rand(9,3,10);
cnet.WC208_16=rand(9,3,10);
cnet.WC208_25=rand(9,3,10);

cnet.WC209_08=rand(9,3,10);
cnet.WC209_17=rand(9,3,10);
cnet.WC209_26=rand(9,3,10);

cnet.WC210_09=rand(9,3,10);
cnet.WC210_18=rand(9,3,10);
cnet.WC210_27=rand(9,3,10);

cnet.biasC201=rand(14,28);
cnet.biasC202=rand(14,28);
cnet.biasC203=rand(12,26);
cnet.biasC204=rand(12,26);
cnet.biasC205=rand(12,26);
cnet.biasC206=rand(12,26);
cnet.biasC207=rand(3,10);
cnet.biasC208=rand(3,10);
cnet.biasC209=rand(3,10);
cnet.biasC210=rand(3,10);

```

```

% layer S2
cnet.WS2C201=0.25*ones(7,14);
cnet.WS2C202=0.25*ones(7,14);
cnet.WS2C203=0.25*ones(6,13);
cnet.WS2C204=0.25*ones(6,13);
cnet.WS2C205=0.25*ones(6,13);
cnet.WS2C206=0.25*ones(6,13);
cnet.WS2C207=ones(3,10);
cnet.WS2C208=ones(3,10);
cnet.WS2C209=ones(3,10);
cnet.WS2C210=ones(3,10);

% layer N1
cnet.WN101=rand(7,14);
cnet.WN102=rand(7,14);
cnet.WN103=rand(6,13);
cnet.WN104=rand(6,13);
cnet.WN105=rand(6,13);
cnet.WN106=rand(6,13);
cnet.WN107=rand(3,10);
cnet.WN108=rand(3,10);
cnet.WN109=rand(3,10);
cnet.WN110=rand(3,10);
cnet.biasN101=rand(1,1);
cnet.biasN102=rand(1,1);
cnet.biasN103=rand(1,1);
cnet.biasN104=rand(1,1);
cnet.biasN105=rand(1,1);
cnet.biasN106=rand(1,1);
cnet.biasN107=rand(1,1);
cnet.biasN108=rand(1,1);
cnet.biasN109=rand(1,1);
cnet.biasN110=rand(1,1);

% layerN2
cnet.WN2=rand(1,10);
cnet.biasN2=rand(1,1);

end

end

end

```

%Структуры свёрточной нейронной сети для локализации текстовых %областей на полутоновых изображениях

```

classdef ConvNN_iBW
    properties
        % Layer C1
        WC101;
        WC102;
        WC103;
        WC104;
        WC105;
        WC106;
        WC107;
        WC108;
        WC109;
        biasC101;
        biasC102;
        biasC103;
        biasC104;
        biasC105;
        biasC106;
        biasC107;
        biasC108;
        biasC109;
    end
end

```

```

% Layer S1
WS1C101; biasS101;
WS1C102; biasS102;
WS1C103; biasS103;
WS1C104; biasS104;
WS1C105; biasS105;
WS1C106; biasS106;
WS1C107; biasS107;
WS1C108; biasS108;
WS1C109; biasS109;

% Layer C2
WC201_01;
WC202_02;
WC202_03;
WC203_04; biasC201;
WC203_05; biasC202;
WC204_06; biasC203;
WC204_07; biasC204;
WC205_08; biasC205;
WC205_09;

% layer S2
WS2C201;
WS2C202;
WS2C203;
WS2C204;
WS2C205;

% layer N1
WN101; biasN101;
WN102; biasN102;
WN103; biasN103;
WN104; biasN104;
WN105; biasN105;

% layerN2
WN2;
biasN2;

end

methods
function cnet=ConvNN_iBW()
% Layer C1

cnet.WC101=rand(25,32,60); cnet.WC108=rand(25,5,12);
cnet.WC102=rand(25,14,28); cnet.WC109=rand(25,5,12);
cnet.WC103=rand(25,14,28);
cnet.WC104=rand(25,14,28); cnet.biasC101=rand(32,60);
cnet.WC105=rand(25,14,28); cnet.biasC102=rand(14,28);
cnet.WC106=rand(25,5,12); cnet.biasC103=rand(14,28);
cnet.WC107=rand(25,5,12); cnet.biasC104=rand(14,28);

```



```

cnet.biasC105=rand(14,28);
cnet.biasC106=rand(5,12);
cnet.biasC107=rand(5,12);

cnet.biasC108=rand(5,12);
cnet.biasC109=rand(5,12);

% Layer S1
cnet.WS1C101=0.25*ones(16,30);
cnet.WS1C102=0.25*ones(12,26);
cnet.WS1C103=0.25*ones(12,26);
cnet.WS1C104=0.25*ones(12,26);
cnet.WS1C105=0.25*ones(12,26);
cnet.WS1C106=0.25*ones(5,12);
cnet.WS1C107=0.25*ones(5,12);
cnet.WS1C108=0.25*ones(5,12);
cnet.WS1C109=0.25*ones(5,12);

cnet.biasS101=rand(16,30);
cnet.biasS102=rand(12,26);
cnet.biasS103=rand(12,26);
cnet.biasS104=rand(12,26);
cnet.biasS105=rand(12,26);
cnet.biasS106=rand(5,12);
cnet.biasS107=rand(5,12);
cnet.biasS108=rand(5,12);
cnet.biasS109=rand(5,12);

% Layer C2
cnet.WC201_01=rand(9,14,28);
cnet.WC202_02=rand(9,12,26);
cnet.WC202_03=rand(9,12,26);
cnet.WC203_04=rand(9,12,26);
cnet.WC203_05=rand(9,12,26);
cnet.WC204_06=rand(9,3,10);
cnet.WC204_07=rand(9,3,10);
cnet.WC205_08=rand(9,3,10);
cnet.WC205_09=rand(9,3,10);

cnet.biasC201=rand(14,28);
cnet.biasC202=rand(12,26);
cnet.biasC203=rand(12,26);
cnet.biasC204=rand(3,10);
cnet.biasC205=rand(3,10);

% layer S2
cnet.WS2C201=0.25*ones(7,14);
cnet.WS2C202=0.25*ones(6,13);
cnet.WS2C203=0.25*ones(6,13);
cnet.WS2C204=ones(3,10);
cnet.WS2C205=ones(3,10);

% layer N1
cnet.WN101=rand(7,14);
cnet.WN102=rand(6,13);
cnet.WN103=rand(6,13);
cnet.WN104=rand(3,10);
cnet.WN105=rand(3,10);

cnet.biasN101=rand(1,1);
cnet.biasN102=rand(1,1);
cnet.biasN103=rand(1,1);
cnet.biasN104=rand(1,1);
cnet.biasN105=rand(1,1);

% layerN2
cnet.WN2=rand(1,10);
cnet.biasN2=rand(1,1);

end

end

end

%Обучение свёрточных нейронных сетей для локализации текстовых

```

%областей на цветных изображениях

```

close all;
clear all;
clc;

% load imdata_label_1836_28Sep.mat;
% err_cnn=zeros(0);
% out_cnn=zeros(1);
% cnet=ConvNN();

%% TRAIN
load train_CNN_ex1836_ep25_rand_28Sep.mat;
len_data=length(imdata);
ep=25;
i1=len_data*(ep-1)+1;
i2=len_data*ep;
max_err=0;
for i=i1:i2
    if err_cnn(i)>max_err
        max_err=err_cnn(i);
    end
end
% circle=length(err_cnn);
% len_data=length(imdata);
% ep=0;
% i1=len_data*(ep-1)+1;
% i2=len_data*ep;
% max_err=2;
while (ep<26) && (max_err>0.01)
    ep=ep+1;
    p=0.1;
    len=length(imdata);
    % for epoch=1:1
        for i=1:length(imdata)
            ir=round(1+(len-1).*rand(1,1));
            circle=circle+1;
            disp(circle);
            [cnet err_cnn out_cnn]=func_backpropagation_ConvNN_tangens_full
                (cnet,imdata,label,ir,p,err_cnn,out_cnn);
        end
    % end
    figure; plot(err_cnn(2:length(err_cnn)));
    i1=1645*(ep-1)+1;
    i2=1645*ep;
    max_err=0;
    for i=i1:i2
        if err_cnn(i)>max_err
            max_err=err_cnn(i);
        end
    end
end
end

```

% func_backpropagation_ConvNN_tangens_full.m

```
function [cnet err_cnn
```

```
out_cnn]=func_backpropagation_ConvNN_tangens_full(cnet,imdata,imlabel,ir,p,err_cnn,out_cnn)
```

```
%% COEFFICIENT MATRIX (WEIGHT, BIAS)
```

```
% Layer C1
```

```
WC101R=cnet.WC101R;          WC110G=cnet.WC110G;          WC119B=cnet.WC119B;
```

```
WC102R=cnet.WC102R;          WC111G=cnet.WC111G;          WC120B=cnet.WC120B;
```

```
WC103R=cnet.WC103R;          WC112G=cnet.WC112G;          WC121B=cnet.WC121B;
```

```
WC104R=cnet.WC104R;          WC113G=cnet.WC113G;          WC122B=cnet.WC122B;
```

```
WC105R=cnet.WC105R;          WC114G=cnet.WC114G;          WC123B=cnet.WC123B;
```

```
WC106R=cnet.WC106R;          WC115G=cnet.WC115G;          WC124B=cnet.WC124B;
```

```
WC107R=cnet.WC107R;          WC116G=cnet.WC116G;          WC125B=cnet.WC125B;
```

```
WC108R=cnet.WC108R;          WC117G=cnet.WC117G;          WC126B=cnet.WC126B;
```

```
WC109R=cnet.WC109R;          WC118G=cnet.WC118G;          WC127B=cnet.WC127B;
```

```
biasC101=cnet.biasC101;      biasC110=cnet.biasC110;      biasC119=cnet.biasC119;
```

```
biasC102=cnet.biasC102;      biasC111=cnet.biasC111;      biasC120=cnet.biasC120;
```

```
biasC103=cnet.biasC103;      biasC112=cnet.biasC112;      biasC121=cnet.biasC121;
```

```
biasC104=cnet.biasC104;      biasC113=cnet.biasC113;      biasC122=cnet.biasC122;
```

```
biasC105=cnet.biasC105;      biasC114=cnet.biasC114;      biasC123=cnet.biasC123;
```

```
biasC106=cnet.biasC106;      biasC115=cnet.biasC115;      biasC124=cnet.biasC124;
```

```
biasC107=cnet.biasC107;      biasC116=cnet.biasC116;      biasC125=cnet.biasC125;
```

```
biasC108=cnet.biasC108;      biasC117=cnet.biasC117;      biasC126=cnet.biasC126;
```

```
biasC109=cnet.biasC109;      biasC118=cnet.biasC118;      biasC127=cnet.biasC127;
```

```
% Layer S1
```

```
WS1C101=cnet.WS1C101;
```

```
biasS101=cnet.biasS101;
```

```
WS1C110=cnet.WS1C110;
```

```
biasS110=cnet.biasS110;
```

```
WS1C119=cnet.WS1C119;
```

```
biasS119=cnet.biasS119;
```

```
% Layer C2
```

```
WC201_01=cnet.WC201_01;
```

```
WC201_19=cnet.WC201_19;
```

```
WC201_10=cnet.WC201_10;
```

```

WC202_01=cnet.WC202_01;
WC202_10=cnet.WC202_10;
WC202_19=cnet.WC202_19;

WC203_02=cnet.WC203_02;
WC203_11=cnet.WC203_11;
WC203_20=cnet.WC203_20;

WC204_03=cnet.WC204_03;
WC204_12=cnet.WC204_12;
WC204_21=cnet.WC204_21;

WC205_04=cnet.WC205_04;
WC205_13=cnet.WC205_13;
WC205_22=cnet.WC205_22;

WC206_05=cnet.WC206_05;
WC206_14=cnet.WC206_14;
WC206_23=cnet.WC206_23;

WC207_06=cnet.WC207_06;
WC207_15=cnet.WC207_15;
WC207_24=cnet.WC207_24;

```

```

WC208_07=cnet.WC208_07;
WC208_16=cnet.WC208_16;
WC208_25=cnet.WC208_25;

WC209_08=cnet.WC209_08;
WC209_17=cnet.WC209_17;
WC209_26=cnet.WC209_26;

WC210_09=cnet.WC210_09;
WC210_18=cnet.WC210_18;
WC210_27=cnet.WC210_27;

biasC201=cnet.biasC201;
biasC202=cnet.biasC202;
biasC203=cnet.biasC203;
biasC204=cnet.biasC204;
biasC205=cnet.biasC205;
biasC206=cnet.biasC206;
biasC207=cnet.biasC207;
biasC208=cnet.biasC208;
biasC209=cnet.biasC209;
biasC210=cnet.biasC210;

```

```
% layer S2
```

```

WS2C201=cnet.WS2C201;
WS2C202=cnet.WS2C202;
WS2C203=cnet.WS2C203;
WS2C204=cnet.WS2C204;
WS2C205=cnet.WS2C205;
WS2C206=cnet.WS2C206;
WS2C207=cnet.WS2C207;
WS2C208=cnet.WS2C208;
WS2C209=cnet.WS2C209;
WS2C210=cnet.WS2C210;

```

```
% layer N1
```

```

WN101=cnet.WN101;
WN102=cnet.WN102;
WN103=cnet.WN103;
WN104=cnet.WN104;
WN105=cnet.WN105;
WN106=cnet.WN106;
WN107=cnet.WN107;
WN108=cnet.WN108;
WN109=cnet.WN109;
WN110=cnet.WN110;

```

```

biasN101=cnet.biasN101;
biasN102=cnet.biasN102;
biasN103=cnet.biasN103;
biasN104=cnet.biasN104;
biasN105=cnet.biasN105;
biasN106=cnet.biasN106;
biasN107=cnet.biasN107;
biasN108=cnet.biasN108;
biasN109=cnet.biasN109;
biasN110=cnet.biasN110;

```

```
% layerN2
```

```

WN2=cnet.WN2;
biasN2=cnet.biasN2;

```

```
%% DIRECT PASSAGE
```

```

label=imlabel(ir);
temp=imdata{ir};
temp=2*im2double(temp)-1;
R=temp(:,:,1);
G=temp(:,:,2);
B=temp(:,:,3);

%% CALCULATE ACTIVATOR OF LAYERS

%% Layer C1
[wR, RA1, RH1, RV1, RD1, RA2, RH2, RV2, RD2] = func_directwavelet_level2( R );
[wG, GA1, GH1, GV1, GD1, GA2, GH2, GV2, GD2] = func_directwavelet_level2( G );
[wB, BA1, BH1, BV1, BD1, BA2, BH2, BV2, BD2] = func_directwavelet_level2( B );

for i=1:32
    for j=1:60
        % chanel R
        wR01(:,i,j)=reshape(wR(i:i+4,j:j+4),1,[]);
        sC101(i,j)=WC101R(:,i,j)'*wR01(:,i,j);

        % chanel G
        wG01(:,i,j)=reshape(wG(i:i+4,j:j+4),1,[]);
        sC110(i,j)=WC110G(:,i,j)'*wG01(:,i,j);

        %chanel B

        wB01(:,i,j)=reshape(wB(i:i+4,j:j+4),1,[]);
        sC119(i,j)=WC119B(:,i,j)'*wB01(:,i,j);

    end
end

for i=1:14
    for j=1:28
        % chanel R
        wR02(:,i,j)=reshape(RA1(i:i+4,j:j+4),1,[]);
        sC102(i,j)=WC102R(:,i,j)'*wR02(:,i,j);

        wR03(:,i,j)=reshape(RH1(i:i+4,j:j+4),1,[]);
        sC103(i,j)=WC103R(:,i,j)'*wR03(:,i,j);

        wR04(:,i,j)=reshape(RV1(i:i+4,j:j+4),1,[]);
        sC104(i,j)=WC104R(:,i,j)'*wR04(:,i,j);

        wR05(:,i,j)=reshape(RD1(i:i+4,j:j+4),1,[]);
        sC105(i,j)=WC105R(:,i,j)'*wR05(:,i,j);

        % chanel G
        wG11(:,i,j)=reshape(GA1(i:i+4,j:j+4),1,[]);
        sC111(i,j)=WC111G(:,i,j)'*wG11(:,i,j);
    end
end

```

```

wG12(:,i,j)=reshape(GH1(i:i+4,j:j+4),1,[]);
sC112(i,j)=WC112G(:,i,j)'*wG12(:,i,j);

wG13(:,i,j)=reshape(GV1(i:i+4,j:j+4),1,[]);
sC113(i,j)=WC113G(:,i,j)'*wG13(:,i,j);

wG14(:,i,j)=reshape(GD1(i:i+4,j:j+4),1,[]);
sC114(i,j)=WC114G(:,i,j)'*wG14(:,i,j);

%chanel B

wB20(:,i,j)=reshape(BA1(i:i+4,j:j+4),1,[]);
sC120(i,j)=WC120B(:,i,j)'*wB20(:,i,j);

wB21(:,i,j)=reshape(BH1(i:i+4,j:j+4),1,[]);
sC121(i,j)=WC121B(:,i,j)'*wB21(:,i,j);

wB22(:,i,j)=reshape(BV1(i:i+4,j:j+4),1,[]);
sC122(i,j)=WC122B(:,i,j)'*wB22(:,i,j);

wB23(:,i,j)=reshape(BD1(i:i+4,j:j+4),1,[]);
sC123(i,j)=WC123B(:,i,j)'*wB23(:,i,j);

end
end

for i=1:5
  for j=1:12

    % chanel R
    wR06(:,i,j)=reshape(RA2(i:i+4,j:j+4),1,[]);
    sC106(i,j)=WC106R(:,i,j)'*wR06(:,i,j);

    wR07(:,i,j)=reshape(RH2(i:i+4,j:j+4),1,[]);
    sC107(i,j)=WC107R(:,i,j)'*wR07(:,i,j);

    wR08(:,i,j)=reshape(RV2(i:i+4,j:j+4),1,[]);
    sC108(i,j)=WC108R(:,i,j)'*wR08(:,i,j);

    wR09(:,i,j)=reshape(RD2(i:i+4,j:j+4),1,[]);
    sC109(i,j)=WC109R(:,i,j)'*wR09(:,i,j);

    % chanel G
    wG15(:,i,j)=reshape(GA2(i:i+4,j:j+4),1,[]);
    sC115(i,j)=WC115G(:,i,j)'*wG15(:,i,j);

    wG16(:,i,j)=reshape(GH2(i:i+4,j:j+4),1,[]);
    sC116(i,j)=WC116G(:,i,j)'*wG16(:,i,j);

    wG17(:,i,j)=reshape(GV2(i:i+4,j:j+4),1,[]);
    sC117(i,j)=WC117G(:,i,j)'*wG17(:,i,j);
  end
end

```

```

wG18(:,i,j)=reshape(GD2(i:i+4,j:j+4),1,[]);
sC118(i,j)=WC118G(:,i,j)'*wG18(:,i,j);

%chanel B

wB24(:,i,j)=reshape(BA2(i:i+4,j:j+4),1,[]);
sC124(i,j)=WC124B(:,i,j)'*wB24(:,i,j);

wB25(:,i,j)=reshape(BH2(i:i+4,j:j+4),1,[]);
sC125(i,j)=WC125B(:,i,j)'*wB25(:,i,j);

wB26(:,i,j)=reshape(BV2(i:i+4,j:j+4),1,[]);
sC126(i,j)=WC126B(:,i,j)'*wB26(:,i,j);

wB27(:,i,j)=reshape(BD2(i:i+4,j:j+4),1,[]);
sC127(i,j)=WC127B(:,i,j)'*wB27(:,i,j);

end
end
sC101=sC101+biasC101;      sC110=sC110+biasC110;      sC119=sC119+biasC119;
sC102=sC102+biasC102;      sC111=sC111+biasC111;      sC120=sC120+biasC120;
sC103=sC103+biasC103;      sC112=sC112+biasC112;      sC121=sC121+biasC121;
sC104=sC104+biasC104;      sC113=sC113+biasC113;      sC122=sC122+biasC122;
sC105=sC105+biasC105;      sC114=sC114+biasC114;      sC123=sC123+biasC123;
sC106=sC106+biasC106;      sC115=sC115+biasC115;      sC124=sC124+biasC124;
sC107=sC107+biasC107;      sC116=sC116+biasC116;      sC125=sC125+biasC125;
sC108=sC108+biasC108;      sC117=sC117+biasC117;      sC126=sC126+biasC126;
sC109=sC109+biasC109;      sC118=sC118+biasC118;      sC127=sC127+biasC127;

uC101=(exp(sC101)-exp(-(sC101)))/(exp(sC101)+exp(-(sC101)));
uC102=(exp(sC102)-exp(-(sC102)))/(exp(sC102)+exp(-(sC102)));
uC103=(exp(sC103)-exp(-(sC103)))/(exp(sC103)+exp(-(sC103)));
uC104=(exp(sC104)-exp(-(sC104)))/(exp(sC104)+exp(-(sC104)));
uC105=(exp(sC105)-exp(-(sC105)))/(exp(sC105)+exp(-(sC105)));
uC106=(exp(sC106)-exp(-(sC106)))/(exp(sC106)+exp(-(sC106)));
uC107=(exp(sC107)-exp(-(sC107)))/(exp(sC107)+exp(-(sC107)));
uC108=(exp(sC108)-exp(-(sC108)))/(exp(sC108)+exp(-(sC108)));
uC109=(exp(sC109)-exp(-(sC109)))/(exp(sC109)+exp(-(sC109)));
uC110=(exp(sC110)-exp(-(sC110)))/(exp(sC110)+exp(-(sC110)));
uC111=(exp(sC111)-exp(-(sC111)))/(exp(sC111)+exp(-(sC111)));
uC112=(exp(sC112)-exp(-(sC112)))/(exp(sC112)+exp(-(sC112)));
uC113=(exp(sC113)-exp(-(sC113)))/(exp(sC113)+exp(-(sC113)));
uC114=(exp(sC114)-exp(-(sC114)))/(exp(sC114)+exp(-(sC114)));
uC115=(exp(sC115)-exp(-(sC115)))/(exp(sC115)+exp(-(sC115)));
uC116=(exp(sC116)-exp(-(sC116)))/(exp(sC116)+exp(-(sC116)));
uC117=(exp(sC117)-exp(-(sC117)))/(exp(sC117)+exp(-(sC117)));
uC118=(exp(sC118)-exp(-(sC118)))/(exp(sC118)+exp(-(sC118)));
uC119=(exp(sC119)-exp(-(sC119)))/(exp(sC119)+exp(-(sC119)));
uC120=(exp(sC120)-exp(-(sC120)))/(exp(sC120)+exp(-(sC120)));
uC121=(exp(sC121)-exp(-(sC121)))/(exp(sC121)+exp(-(sC121)));
uC122=(exp(sC122)-exp(-(sC122)))/(exp(sC122)+exp(-(sC122)));
uC123=(exp(sC123)-exp(-(sC123)))/(exp(sC123)+exp(-(sC123)));
uC124=(exp(sC124)-exp(-(sC124)))/(exp(sC124)+exp(-(sC124)));
uC125=(exp(sC125)-exp(-(sC125)))/(exp(sC125)+exp(-(sC125)));
uC126=(exp(sC126)-exp(-(sC126)))/(exp(sC126)+exp(-(sC126)));

```

```

uC127=(exp(sC127)-exp(-(sC127)))/(exp(sC127)+exp(-(sC127)));

%% Layer S1

for i=1:16
    i1=2*(i-1)+1;
    i2=i1+1;
    for j=1:30
        j1=2*(j-1)+1;
        j2=j1+1;
        uS101(i,j)=0.25*(sum(sum(uC101(i1:i2,j1:j2))));
        uS110(i,j)=0.25*(sum(sum(uC110(i1:i2,j1:j2))));
        uS119(i,j)=0.25*(sum(sum(uC119(i1:i2,j1:j2))));
    end
end
uS102=uC102;          uS111=uC111;          uS120=uC120;
uS103=uC103;          uS112=uC112;          uS121=uC121;
uS104=uC104;          uS113=uC113;          uS122=uC122;
uS105=uC105;          uS114=uC114;          uS123=uC123;
uS106=uC106;          uS115=uC115;          uS124=uC124;
uS107=uC107;          uS116=uC116;          uS125=uC125;
uS108=uC108;          uS117=uC117;          uS126=uC126;
uS109=uC109;          uS118=uC118;          uS127=uC127;

%% Layer C2

for i=1:14
    for j=1:28

        S101C2(:,i,j)=reshape(uS101(i:i+2,j:j+2),1,[]);
        S110C2(:,i,j)=reshape(uS110(i:i+2,j:j+2),1,[]);
        S119C2(:,i,j)=reshape(uS119(i:i+2,j:j+2),1,[]);

        S101C2ij=S101C2(:,i,j);
        S110C2ij=S110C2(:,i,j);
        S119C2ij=S119C2(:,i,j);

        %sC201
        WC201_1ij=WC201_01(:,i,j);
        WC201_10ij=WC201_10(:,i,j);
        WC201_19ij=WC201_19(:,i,j);

sC201(i,j)=WC201_1ij'*S101C2ij+WC201_10ij'*S110C2ij+WC201_19ij'*S119C2ij;
        %sC202
        WC202_1ij=WC202_01(:,i,j);
        WC202_10ij=WC202_10(:,i,j);
        WC202_19ij=WC202_19(:,i,j);
        sC202(i,j)=WC202_1ij'*S101C2ij+WC202_10ij'*S110C2ij+WC202_19ij'*S119C2ij;

    end
end

for i=1:12
    for j=1:26

```



```

S102C2(:,i,j)=reshape(uS102(i:i+2,j:j+2),1,[]);
S103C2(:,i,j)=reshape(uS103(i:i+2,j:j+2),1,[]);
S104C2(:,i,j)=reshape(uS104(i:i+2,j:j+2),1,[]);
S105C2(:,i,j)=reshape(uS105(i:i+2,j:j+2),1,[]);

S111C2(:,i,j)=reshape(uS111(i:i+2,j:j+2),1,[]);
S112C2(:,i,j)=reshape(uS112(i:i+2,j:j+2),1,[]);
S113C2(:,i,j)=reshape(uS113(i:i+2,j:j+2),1,[]);
S114C2(:,i,j)=reshape(uS114(i:i+2,j:j+2),1,[]);

S120C2(:,i,j)=reshape(uS120(i:i+2,j:j+2),1,[]);
S121C2(:,i,j)=reshape(uS121(i:i+2,j:j+2),1,[]);
S122C2(:,i,j)=reshape(uS122(i:i+2,j:j+2),1,[]);
S123C2(:,i,j)=reshape(uS123(i:i+2,j:j+2),1,[]);

%sC203
S102C2ij=S102C2(:,i,j);
S111C2ij=S111C2(:,i,j);
S120C2ij=S120C2(:,i,j);

WC203_2ij=WC203_02(:,i,j);
WC203_11ij=WC203_11(:,i,j);
WC203_20ij=WC203_20(:,i,j);

sC203(i,j)=WC203_2ij'*S102C2ij+WC203_11ij'*S111C2ij+WC203_20ij'*S120C2ij;

%sC204
S103C2ij=S103C2(:,i,j);
S112C2ij=S112C2(:,i,j);
S121C2ij=S121C2(:,i,j);

WC204_3ij=WC204_03(:,i,j);
WC204_12ij=WC204_12(:,i,j);
WC204_21ij=WC204_21(:,i,j);

sC204(i,j)=WC204_3ij'*S103C2ij+WC204_12ij'*S112C2ij+WC204_21ij'*S121C2ij;

%sC205
S104C2ij=S104C2(:,i,j);
S113C2ij=S113C2(:,i,j);
S122C2ij=S122C2(:,i,j);

WC205_4ij=WC205_04(:,i,j);
WC205_13ij=WC205_13(:,i,j);
WC205_22ij=WC205_22(:,i,j);

sC205(i,j)=WC205_4ij'*S104C2ij+WC205_13ij'*S113C2ij+WC205_22ij'*S122C2ij;

%sC206
S105C2ij=S105C2(:,i,j);
S114C2ij=S114C2(:,i,j);
S123C2ij=S123C2(:,i,j);

WC206_5ij=WC206_05(:,i,j);

```

```

        WC206_14ij=WC206_14(:,i,j);
        WC206_23ij=WC206_23(:,i,j);

sC206(i,j)=WC206_5ij'*S105C2ij+WC206_14ij'*S114C2ij+WC206_23ij'*S123C2ij;

    end
end
for i=1:3
    for j=1:10

        S106C2(:,i,j)=reshape(uS106(i:i+2,j:j+2),1,[]);
        S107C2(:,i,j)=reshape(uS107(i:i+2,j:j+2),1,[]);
        S108C2(:,i,j)=reshape(uS108(i:i+2,j:j+2),1,[]);
        S109C2(:,i,j)=reshape(uS109(i:i+2,j:j+2),1,[]);

        S115C2(:,i,j)=reshape(uS115(i:i+2,j:j+2),1,[]);
        S116C2(:,i,j)=reshape(uS116(i:i+2,j:j+2),1,[]);
        S117C2(:,i,j)=reshape(uS117(i:i+2,j:j+2),1,[]);
        S118C2(:,i,j)=reshape(uS118(i:i+2,j:j+2),1,[]);

        S124C2(:,i,j)=reshape(uS124(i:i+2,j:j+2),1,[]);
        S125C2(:,i,j)=reshape(uS125(i:i+2,j:j+2),1,[]);
        S126C2(:,i,j)=reshape(uS126(i:i+2,j:j+2),1,[]);
        S127C2(:,i,j)=reshape(uS127(i:i+2,j:j+2),1,[]);

    %sC207

        S106C2ij=S106C2(:,i,j);
        S115C2ij=S115C2(:,i,j);
        S124C2ij=S124C2(:,i,j);

        WC207_6ij=WC207_06(:,i,j);
        WC207_15ij=WC207_15(:,i,j);
        WC207_24ij=WC207_24(:,i,j);

sC207(i,j)=WC207_6ij'*S106C2ij+WC207_15ij'*S115C2ij+WC207_24ij'*S124C2ij;

    %sC208

        S107C2ij=S107C2(:,i,j);
        S116C2ij=S116C2(:,i,j);
        S125C2ij=S125C2(:,i,j);

        WC208_7ij=WC208_07(:,i,j);
        WC208_16ij=WC208_16(:,i,j);
        WC208_25ij=WC208_25(:,i,j);

sC208(i,j)=WC208_7ij'*S107C2ij+WC208_16ij'*S116C2ij+WC208_25ij'*S125C2ij;

    %sC209

        S108C2ij=S108C2(:,i,j);
        S117C2ij=S117C2(:,i,j);
        S126C2ij=S126C2(:,i,j);

        WC209_8ij=WC209_08(:,i,j);
        WC209_17ij=WC209_17(:,i,j);

```

```

WC209_26ij=WC209_26(:,i,j);

sC209(i,j)=WC209_8ij'*S108C2ij+WC209_17ij'*S117C2ij+WC209_26ij'*S126C2ij;

%SC210
S109C2ij=S109C2(:,i,j);
S118C2ij=S118C2(:,i,j);
S127C2ij=S127C2(:,i,j);

WC210_9ij=WC210_09(:,i,j);
WC210_18ij=WC210_18(:,i,j);
WC210_27ij=WC210_27(:,i,j);

sC210(i,j)=WC210_9ij'*S109C2ij+WC210_18ij'*S118C2ij+WC210_27ij'*S127C2ij;

    end
end

%uC201
sC201=sC201+biasC201;
uC201=(exp(sC201)-exp(-(sC201)))/(exp(sC201)+exp(-(sC201)));
%uC202
sC202=sC202+biasC202;
uC202=(exp(sC202)-exp(-(sC202)))/(exp(sC202)+exp(-(sC202)));
%uC203
sC203=sC203+biasC203;
uC203=(exp(sC203)-exp(-(sC203)))/(exp(sC203)+exp(-(sC203)));
%uC204
sC204=sC204+biasC204;
uC204=(exp(sC204)-exp(-(sC204)))/(exp(sC204)+exp(-(sC204)));
%uC205
sC205=sC205+biasC205;
uC205=(exp(sC205)-exp(-(sC205)))/(exp(sC205)+exp(-(sC205)));
%uC206
sC206=sC206+biasC206;
uC206=(exp(sC206)-exp(-(sC206)))/(exp(sC206)+exp(-(sC206)));
%uC207
sC207=sC207+biasC207;
uC207=(exp(sC207)-exp(-(sC207)))/(exp(sC207)+exp(-(sC207)));
%uC208
sC208=sC208+biasC208;
uC208=(exp(sC208)-exp(-(sC208)))/(exp(sC208)+exp(-(sC208)));
%uC209
sC209=sC209+biasC209;
uC209=(exp(sC209)-exp(-(sC209)))/(exp(sC209)+exp(-(sC209)));
%uC210
sC210=sC210+biasC210;
uC210=(exp(sC210)-exp(-(sC210)))/(exp(sC210)+exp(-(sC210)));

%% layer S2
for i=1:7
    i1=2*(i-1)+1;
    i2=i1+1;
    for j=1:14
        j1=2*(j-1)+1;

```

```

        j2=j1+1;
        uS201(i,j)=0.25*(sum(sum(uC201(i1:i2,j1:j2))));
        uS202(i,j)=0.25*(sum(sum(uC202(i1:i2,j1:j2))));

    end
end
for i=1:6
    i1=2*(i-1)+1;
    i2=i1+1;
    for j=1:13
        j1=2*(j-1)+1;
        j2=j1+1;
        uS203(i,j)=0.25*(sum(sum(uC203(i1:i2,j1:j2))));
        uS204(i,j)=0.25*(sum(sum(uC204(i1:i2,j1:j2))));
        uS205(i,j)=0.25*(sum(sum(uC205(i1:i2,j1:j2))));
        uS206(i,j)=0.25*(sum(sum(uC206(i1:i2,j1:j2))));
    end
end
end
uS207=uC207;
uS208=uC208;
uS209=uC209;
uS210=uC210;

%% layer N1
sN101=sum(sum(WN101.*uS201))/98+biasN101;
sN102=sum(sum(WN102.*uS202))/98+biasN102;
sN103=sum(sum(WN103.*uS203))/98+biasN103;
sN104=sum(sum(WN104.*uS204))/98+biasN104;
sN105=sum(sum(WN105.*uS205))/98+biasN105;
sN106=sum(sum(WN106.*uS206))/98+biasN106;
sN107=sum(sum(WN107.*uS207))/98+biasN107;
sN108=sum(sum(WN108.*uS208))/98+biasN108;
sN109=sum(sum(WN109.*uS209))/98+biasN109;
sN110=sum(sum(WN110.*uS210))/98+biasN110;

uN101=(exp(sN101)-exp(-(sN101)))/(exp(sN101)+exp(-(sN101)));
uN102=(exp(sN102)-exp(-(sN102)))/(exp(sN102)+exp(-(sN102)));
uN103=(exp(sN103)-exp(-(sN103)))/(exp(sN103)+exp(-(sN103)));
uN104=(exp(sN104)-exp(-(sN104)))/(exp(sN104)+exp(-(sN104)));
uN105=(exp(sN105)-exp(-(sN105)))/(exp(sN105)+exp(-(sN105)));
uN106=(exp(sN106)-exp(-(sN106)))/(exp(sN106)+exp(-(sN106)));
uN107=(exp(sN107)-exp(-(sN107)))/(exp(sN107)+exp(-(sN107)));
uN108=(exp(sN108)-exp(-(sN108)))/(exp(sN108)+exp(-(sN108)));
uN109=(exp(sN109)-exp(-(sN109)))/(exp(sN109)+exp(-(sN109)));
uN110=(exp(sN110)-exp(-(sN110)))/(exp(sN110)+exp(-(sN110)));

%% layer N2
tam1=WN2(1,1)*uN101+WN2(1,2)*uN102+WN2(1,3)*uN103+WN2(1,4)*uN104+WN2(1,5)*uN105;
tam2=WN2(1,6)*uN106+WN2(1,7)*uN107+WN2(1,8)*uN108+WN2(1,9)*uN109+WN2(1,10)*uN110;
;
sN2=tam1+tam2+biasN2;
uN2=(exp(sN2)-exp(-sN2))/(exp(sN2)+exp(-sN2));
out=uN2;
out_cnn=[out_cnn out];

```

```

%% CALCULATE ERRORS OF LAYERS
C=label;
e=0.5*((C-out)^2);
err_cnn=[err_cnn e];
%% layer N2
errN2=(C-out)*(1-out)*(1+out)/2;

%% layer N1
errN101=0.5*(1+uN101)*(1-uN101)*(errN2*WN2(1,1));
errN102=0.5*(1+uN102)*(1-uN102)*(errN2*WN2(1,2));
errN103=0.5*(1+uN103)*(1-uN103)*(errN2*WN2(1,3));
errN104=0.5*(1+uN104)*(1-uN104)*(errN2*WN2(1,4));
errN105=0.5*(1+uN105)*(1-uN105)*(errN2*WN2(1,5));
errN106=0.5*(1+uN106)*(1-uN106)*(errN2*WN2(1,6));
errN107=0.5*(1+uN107)*(1-uN107)*(errN2*WN2(1,7));
errN108=0.5*(1+uN108)*(1-uN108)*(errN2*WN2(1,8));
errN109=0.5*(1+uN109)*(1-uN109)*(errN2*WN2(1,9));
errN110=0.5*(1+uN110)*(1-uN110)*(errN2*WN2(1,10));

%% layer S2
errS201=0.5*(1+uS201).*(1-uS201).*(errN101*WN101);
errS202=0.5*(1+uS202).*(1-uS202).*(errN102*WN102);
errS203=0.5*(1+uS203).*(1-uS203).*(errN103*WN103);
errS204=0.5*(1+uS204).*(1-uS204).*(errN104*WN104);
errS205=0.5*(1+uS205).*(1-uS205).*(errN105*WN105);
errS206=0.5*(1+uS206).*(1-uS206).*(errN106*WN106);
errS207=0.5*(1+uS207).*(1-uS207).*(errN107*WN107);
errS208=0.5*(1+uS208).*(1-uS208).*(errN108*WN108);
errS209=0.5*(1+uS209).*(1-uS209).*(errN109*WN109);
errS210=0.5*(1+uS210).*(1-uS210).*(errN110*WN110);

%% layer C2
for i=1:7
    i1=2*(i-1)+1;
    i2=i1+1;
    for j=1:14
        j1=2*(j-1)+1;
        j2=j1+1;
        errC201(i1:i2,j1:j2)=errS201(i,j).*WS2C201(i,j);
        errC202(i1:i2,j1:j2)=errS202(i,j).*WS2C202(i,j);
    end
end
for i=1:6
    i1=2*(i-1)+1;
    i2=i1+1;
    for j=1:13
        j1=2*(j-1)+1;
        j2=j1+1;
        errC203(i1:i2,j1:j2)=errS203(i,j).*WS2C203(i,j);
        errC204(i1:i2,j1:j2)=errS204(i,j).*WS2C204(i,j);
        errC205(i1:i2,j1:j2)=errS205(i,j).*WS2C205(i,j);
        errC206(i1:i2,j1:j2)=errS206(i,j).*WS2C206(i,j);
    end
end
end

```

```

errC207=errS207;
errC208=errS208;
errC209=errS209;
errC210=errS210;

%% Layer S1
rWC201_01=zeros(16,30);
rWC201_10=zeros(16,30);
rWC201_19=zeros(16,30);
rWC202_01=zeros(16,30);
rWC202_10=zeros(16,30);
rWC202_19=zeros(16,30);

for i=1:14
    for j=1:28

        w=errC201(i,j)*WC201_01(:,i,j);
        ww=reshape(w,3,3);
        rWC201_01(i:i+2,j:j+2)=rWC201_01(i:i+2,j:j+2)+ww;

        w=errC201(i,j)*WC201_10(:,i,j);
        ww=reshape(w,3,3);
        rWC201_10(i:i+2,j:j+2)=rWC201_10(i:i+2,j:j+2)+ww;

        w=errC201(i,j)*WC201_19(:,i,j);
        ww=reshape(w,3,3);
        rWC201_19(i:i+2,j:j+2)=rWC201_19(i:i+2,j:j+2)+ww;

        w=errC202(i,j)*WC202_01(:,i,j);
        ww=reshape(w,3,3);
        rWC202_01(i:i+2,j:j+2)=rWC202_01(i:i+2,j:j+2)+ww;

        w=errC202(i,j)*WC202_10(:,i,j);
        ww=reshape(w,3,3);
        rWC202_10(i:i+2,j:j+2)=rWC202_10(i:i+2,j:j+2)+ww;

        w=errC202(i,j)*WC202_19(:,i,j);
        ww=reshape(w,3,3);
        rWC202_19(i:i+2,j:j+2)=rWC202_19(i:i+2,j:j+2)+ww;

    end
end
errS101=0.5*(1+uS101).*(1-uS101).*(rWC201_01+rWC202_01);
errS110=0.5*(1+uS110).*(1-uS110).*(rWC201_10+rWC202_10);
errS119=0.5*(1+uS119).*(1-uS119).*(rWC201_19+rWC202_19);

rWC203_02=zeros(14,28);
rWC203_11=zeros(14,28);
rWC203_20=zeros(14,28);

rWC204_03=zeros(14,28);

```

```

rWC204_12=zeros(14,28);
rWC204_21=zeros(14,28);

rWC205_04=zeros(14,28);
rWC205_13=zeros(14,28);
rWC205_22=zeros(14,28);

rWC206_05=zeros(14,28);
rWC206_14=zeros(14,28);
rWC206_23=zeros(14,28);

for i=1:12
    ii=i;
    for j=1:26
        jj=j;

        w=errC203(i,j)*WC203_02(:,i,j);
        ww=reshape(w,3,3);
        rWC203_02(ii:ii+2,jj:jj+2)=rWC203_02(ii:ii+2,jj:jj+2)+ww;

        w=errC203(i,j)*WC203_11(:,i,j);
        ww=reshape(w,3,3);
        rWC203_11(ii:ii+2,jj:jj+2)=rWC203_11(ii:ii+2,jj:jj+2)+ww;

        w=errC203(i,j)*WC203_20(:,i,j);
        ww=reshape(w,3,3);
        rWC203_20(ii:ii+2,jj:jj+2)=rWC203_20(ii:ii+2,jj:jj+2)+ww;

        w=errC204(i,j)*WC204_03(:,i,j);
        ww=reshape(w,3,3);
        rWC204_03(ii:ii+2,jj:jj+2)=rWC204_03(ii:ii+2,jj:jj+2)+ww;

        w=errC204(i,j)*WC204_12(:,i,j);
        ww=reshape(w,3,3);
        rWC204_12(ii:ii+2,jj:jj+2)=rWC204_12(ii:ii+2,jj:jj+2)+ww;

        w=errC204(i,j)*WC204_21(:,i,j);
        ww=reshape(w,3,3);
        rWC204_21(ii:ii+2,jj:jj+2)=rWC204_21(ii:ii+2,jj:jj+2)+ww;

        w=errC205(i,j)*WC205_04(:,i,j);
        ww=reshape(w,3,3);
        rWC205_04(ii:ii+2,jj:jj+2)=rWC205_04(ii:ii+2,jj:jj+2)+ww;

        w=errC205(i,j)*WC205_13(:,i,j);
        ww=reshape(w,3,3);
        rWC205_13(ii:ii+2,jj:jj+2)=rWC205_13(ii:ii+2,jj:jj+2)+ww;

        w=errC205(i,j)*WC205_22(:,i,j);
        ww=reshape(w,3,3);
        rWC205_22(ii:ii+2,jj:jj+2)=rWC205_22(ii:ii+2,jj:jj+2)+ww;

        w=errC206(i,j)*WC206_05(:,i,j);

```

```

ww=reshape(w,3,3);
rWC206_05(ii:ii+2,jj:jj+2)=rWC206_05(ii:ii+2,jj:jj+2)+ww;

w=errC206(i,j)*WC206_14(:,i,j);
ww=reshape(w,3,3);
rWC206_14(ii:ii+2,jj:jj+2)=rWC206_14(ii:ii+2,jj:jj+2)+ww;

w=errC206(i,j)*WC206_23(:,i,j);
ww=reshape(w,3,3);
rWC206_23(ii:ii+2,jj:jj+2)=rWC206_23(ii:ii+2,jj:jj+2)+ww;

end
end
errS102=0.5*(1+uS102).*(1-uS102).*rWC203_02;
errS103=0.5*(1+uS103).*(1-uS103).*rWC204_03;
errS104=0.5*(1+uS104).*(1-uS104).*rWC205_04;
errS105=0.5*(1+uS105).*(1-uS105).*rWC206_05;

errS111=0.5*(1+uS111).*(1-uS111).*rWC203_11;
errS112=0.5*(1+uS112).*(1-uS112).*rWC204_12;
errS113=0.5*(1+uS113).*(1-uS113).*rWC205_13;
errS114=0.5*(1+uS114).*(1-uS114).*rWC206_14;

errS120=0.5*(1+uS120).*(1-uS120).*rWC203_20;
errS121=0.5*(1+uS121).*(1-uS121).*rWC204_21;
errS122=0.5*(1+uS122).*(1-uS122).*rWC205_22;
errS123=0.5*(1+uS123).*(1-uS123).*rWC206_23;

rWC207_06=zeros(5,12);
rWC207_15=zeros(5,12);
rWC207_24=zeros(5,12);

rWC208_07=zeros(5,12);
rWC208_16=zeros(5,12);
rWC208_25=zeros(5,12);

rWC209_08=zeros(5,12);
rWC209_17=zeros(5,12);
rWC209_26=zeros(5,12);

rWC210_09=zeros(5,12);
rWC210_18=zeros(5,12);
rWC210_27=zeros(5,12);

for i=1:3
    ii=i;
    for j=1:10
        jj=j;

        w=errC207(i,j)*WC207_06(:,i,j);
        ww=reshape(w,3,3);
        rWC207_06(ii:ii+2,jj:jj+2)=rWC207_06(ii:ii+2,jj:jj+2)+ww;

```



```

w=errC207(i,j)*WC207_15(:,i,j);
ww=reshape(w,3,3);
rWC207_15(ii:ii+2,jj:jj+2)=rWC207_15(ii:ii+2,jj:jj+2)+ww;

w=errC207(i,j)*WC207_24(:,i,j);
ww=reshape(w,3,3);
rWC207_24(ii:ii+2,jj:jj+2)=rWC207_24(ii:ii+2,jj:jj+2)+ww;

w=errC208(i,j)*WC208_07(:,i,j);
ww=reshape(w,3,3);
rWC208_07(ii:ii+2,jj:jj+2)=rWC208_07(ii:ii+2,jj:jj+2)+ww;

w=errC208(i,j)*WC208_16(:,i,j);
ww=reshape(w,3,3);
rWC208_16(ii:ii+2,jj:jj+2)=rWC208_16(ii:ii+2,jj:jj+2)+ww;

w=errC208(i,j)*WC208_25(:,i,j);
ww=reshape(w,3,3);
rWC208_25(ii:ii+2,jj:jj+2)=rWC208_25(ii:ii+2,jj:jj+2)+ww;

w=errC209(i,j)*WC209_08(:,i,j);
ww=reshape(w,3,3);
rWC209_08(ii:ii+2,jj:jj+2)=rWC209_08(ii:ii+2,jj:jj+2)+ww;

w=errC209(i,j)*WC209_17(:,i,j);
ww=reshape(w,3,3);
rWC209_17(ii:ii+2,jj:jj+2)=rWC209_17(ii:ii+2,jj:jj+2)+ww;

w=errC209(i,j)*WC209_26(:,i,j);
ww=reshape(w,3,3);
rWC209_26(ii:ii+2,jj:jj+2)=rWC209_26(ii:ii+2,jj:jj+2)+ww;

w=errC210(i,j)*WC210_09(:,i,j);
ww=reshape(w,3,3);
rWC210_09(ii:ii+2,jj:jj+2)=rWC210_09(ii:ii+2,jj:jj+2)+ww;

w=errC210(i,j)*WC210_18(:,i,j);
ww=reshape(w,3,3);
rWC210_18(ii:ii+2,jj:jj+2)=rWC210_18(ii:ii+2,jj:jj+2)+ww;

w=errC210(i,j)*WC210_27(:,i,j);
ww=reshape(w,3,3);
rWC210_27(ii:ii+2,jj:jj+2)=rWC210_27(ii:ii+2,jj:jj+2)+ww;

end

errS106=0.5*(1+uS106).* (1-uS106). *rWC207_06;
errS107=0.5*(1+uS107). * (1-uS107). *rWC208_07;
errS108=0.5*(1+uS108). * (1-uS108). *rWC209_08;
errS109=0.5*(1+uS109). * (1-uS109). *rWC210_09;

```

```

errS115=0.5*(1+uS115).* (1-uS115) .*rWC207_15;
errS116=0.5*(1+uS116) .* (1-uS116) .*rWC208_16;
errS117=0.5*(1+uS117) .* (1-uS117) .*rWC209_17;
errS118=0.5*(1+uS118) .* (1-uS118) .*rWC210_18;

errS124=0.5*(1+uS124) .* (1-uS124) .*rWC207_24;
errS125=0.5*(1+uS125) .* (1-uS125) .*rWC208_25;
errS126=0.5*(1+uS126) .* (1-uS126) .*rWC209_26;
errS127=0.5*(1+uS127) .* (1-uS127) .*rWC210_27;

%% layer C1
for i=1:16
    i1=2*(i-1)+1;
    i2=i1+1;
    for j=1:30
        j1=2*(j-1)+1;
        j2=j1+1;
        errC101(i1:i2,j1:j2)=errS101(i,j) .*WS1C101(i,j);
        errC110(i1:i2,j1:j2)=errS110(i,j) .*WS1C110(i,j);
        errC119(i1:i2,j1:j2)=errS119(i,j) .*WS1C119(i,j);
    end
end
errC102=errS102;
errC103=errS103;
errC104=errS104;
errC105=errS105;
errC106=errS106;
errC107=errS107;
errC108=errS108;
errC109=errS109;

errC111=errS111;
errC112=errS112;
errC113=errS113;
errC114=errS114;
errC115=errS115;
errC116=errS116;
errC117=errS117;
errC118=errS118;

errC120=errS120;
errC121=errS121;
errC122=errS122;
errC123=errS123;
errC124=errS124;
errC125=errS125;
errC126=errS126;
errC127=errS127;

%% CHANGE WEIGHT AND BIAS
% layer N2
newWN2(1,1)=WN2(1,1)+p*errN2*uN101;
newWN2(1,2)=WN2(1,2)+p*errN2*uN102;
newWN2(1,3)=WN2(1,3)+p*errN2*uN103;
newWN2(1,4)=WN2(1,4)+p*errN2*uN104;

```

```

newWN2(1,5)=WN2(1,5)+p*errN2*uN105;
newWN2(1,6)=WN2(1,6)+p*errN2*uN106;
newWN2(1,7)=WN2(1,7)+p*errN2*uN107;
newWN2(1,8)=WN2(1,8)+p*errN2*uN108;
newWN2(1,9)=WN2(1,9)+p*errN2*uN109;
newWN2(1,10)=WN2(1,10)+p*errN2*uN110;

newbiasN2=biasN2+p*errN2;

%% layer N1
newWN101=WN101+p*errN101*uS201;
newWN102=WN102+p*errN102*uS202;
newWN103=WN103+p*errN103*uS203;
newWN104=WN104+p*errN104*uS204;
newWN105=WN105+p*errN105*uS205;
newWN106=WN106+p*errN106*uS206;
newWN107=WN107+p*errN107*uS207;
newWN108=WN108+p*errN108*uS208;
newWN109=WN109+p*errN109*uS209;
newWN110=WN110+p*errN110*uS210;

newbiasN101=biasN101+p*errN101;
newbiasN102=biasN102+p*errN102;
newbiasN103=biasN103+p*errN103;
newbiasN104=biasN104+p*errN104;
newbiasN105=biasN105+p*errN105;
newbiasN106=biasN106+p*errN106;
newbiasN107=biasN107+p*errN107;
newbiasN108=biasN108+p*errN108;
newbiasN109=biasN109+p*errN109;
newbiasN110=biasN110+p*errN110;

%% layer C2

for i=1:14
    for j=1:28

        S101C2ij=reshape(S101C2(:,i,j),3,3);
        S110C2ij=reshape(S110C2(:,i,j),3,3);
        S119C2ij=reshape(S119C2(:,i,j),3,3);

% C201 - WC201_01, WC201_10, WC201_19
        errC201ij=errC201(i,j);

        WC201_01ij=reshape(WC201_01(:,i,j),3,3);
        newWC201_01ij=WC201_01ij+p*errC201ij*S101C2ij;
        newWC201_01(:,i,j)=reshape(newWC201_01ij,1,[]);

        WC201_10ij=reshape(WC201_10(:,i,j),3,3);
        newWC201_10ij=WC201_10ij+p*errC201ij*S110C2ij;
        newWC201_10(:,i,j)=reshape(newWC201_10ij,1,[]);

        WC201_19ij=reshape(WC201_19(:,i,j),3,3);
        newWC201_19ij=WC201_19ij+p*errC201ij*S119C2ij;

```

```

    newWC201_19(:,i,j)=reshape(newWC201_19ij,1,[]);

% C202 - WC201_01, WC201_10, WC201_19
    errC202ij=errC202(i,j);

    WC202_01ij=reshape(WC202_01(:,i,j),3,3);
    newWC202_01ij=WC202_01ij+p*errC202ij*S101C2ij;
    newWC202_01(:,i,j)=reshape(newWC202_01ij,1,[]);

    WC202_10ij=reshape(WC202_10(:,i,j),3,3);
    newWC202_10ij=WC202_10ij+p*errC202ij*S110C2ij;
    newWC202_10(:,i,j)=reshape(newWC202_10ij,1,[]);

    WC202_19ij=reshape(WC202_19(:,i,j),3,3);
    newWC202_19ij=WC202_19ij+p*errC202ij*S119C2ij;
    newWC202_19(:,i,j)=reshape(newWC202_19ij,1,[]);

end
end

for i=1:12
    for j=1:26

        S102C2ij=reshape(S102C2(:,i,j),3,3);
        S103C2ij=reshape(S103C2(:,i,j),3,3);
        S104C2ij=reshape(S104C2(:,i,j),3,3);
        S105C2ij=reshape(S105C2(:,i,j),3,3);

        S111C2ij=reshape(S111C2(:,i,j),3,3);
        S112C2ij=reshape(S112C2(:,i,j),3,3);
        S113C2ij=reshape(S113C2(:,i,j),3,3);
        S114C2ij=reshape(S114C2(:,i,j),3,3);

        S120C2ij=reshape(S120C2(:,i,j),3,3);
        S121C2ij=reshape(S121C2(:,i,j),3,3);
        S122C2ij=reshape(S122C2(:,i,j),3,3);
        S123C2ij=reshape(S123C2(:,i,j),3,3);

% C203 - WC203_02, WC203_11, WC203_20
        errC203ij=errC203(i,j);

        WC203_02ij=reshape(WC203_02(:,i,j),3,3);
        newWC203_02ij=WC203_02ij+p*errC203ij*S102C2ij;
        newWC203_02(:,i,j)=reshape(newWC203_02ij,1,[]);

        WC203_11ij=reshape(WC203_11(:,i,j),3,3);
        newWC203_11ij=WC203_11ij+p*errC203ij*S111C2ij;
        newWC203_11(:,i,j)=reshape(newWC203_11ij,1,[]);

        WC203_20ij=reshape(WC203_20(:,i,j),3,3);
        newWC203_20ij=WC203_20ij+p*errC203ij*S120C2ij;
        newWC203_20(:,i,j)=reshape(newWC203_20ij,1,[]);

% C204 - WC204_03, WC204_12, WC204_21

```

```

errC204ij=errC204(i,j);

WC204_03ij=reshape(WC204_03(:,i,j),3,3);
newWC204_03ij=WC204_03ij+p*errC204ij*S103C2ij;
newWC204_03(:,i,j)=reshape(newWC204_03ij,1,[]);

WC204_12ij=reshape(WC204_12(:,i,j),3,3);
newWC204_12ij=WC204_12ij+p*errC204ij*S112C2ij;
newWC204_12(:,i,j)=reshape(newWC204_12ij,1,[]);

WC204_21ij=reshape(WC204_21(:,i,j),3,3);
newWC204_21ij=WC204_21ij+p*errC204ij*S121C2ij;
newWC204_21(:,i,j)=reshape(newWC204_21ij,1,[]);

% C205 - WC205_04, WC205_13, WC205_22
errC205ij=errC205(i,j);

WC205_04ij=reshape(WC205_04(:,i,j),3,3);
newWC205_04ij=WC205_04ij+p*errC205ij*S104C2ij;
newWC205_04(:,i,j)=reshape(newWC205_04ij,1,[]);

WC205_13ij=reshape(WC205_13(:,i,j),3,3);
newWC205_13ij=WC205_13ij+p*errC205ij*S113C2ij;
newWC205_13(:,i,j)=reshape(newWC205_13ij,1,[]);

WC205_22ij=reshape(WC205_22(:,i,j),3,3);
newWC205_22ij=WC205_22ij+p*errC205ij*S122C2ij;
newWC205_22(:,i,j)=reshape(newWC205_22ij,1,[]);

% C206 - WC206_05, WC206_14, WC206_23
errC206ij=errC206(i,j);

WC206_05ij=reshape(WC206_05(:,i,j),3,3);
newWC206_05ij=WC206_05ij+p*errC206ij*S105C2ij;
newWC206_05(:,i,j)=reshape(newWC206_05ij,1,[]);

WC206_14ij=reshape(WC206_14(:,i,j),3,3);
newWC206_14ij=WC206_14ij+p*errC206ij*S114C2ij;
newWC206_14(:,i,j)=reshape(newWC206_14ij,1,[]);

WC206_23ij=reshape(WC206_23(:,i,j),3,3);
newWC206_23ij=WC206_23ij+p*errC206ij*S123C2ij;
newWC206_23(:,i,j)=reshape(newWC206_23ij,1,[]);

end
end

for i=1:3
    for j=1:10

        S106C2ij=reshape(S106C2(:,i,j),3,3);
        S107C2ij=reshape(S107C2(:,i,j),3,3);
        S108C2ij=reshape(S108C2(:,i,j),3,3);
        S109C2ij=reshape(S109C2(:,i,j),3,3);
    
```

```

S115C2ij=reshape(S115C2(:,i,j),3,3);
S116C2ij=reshape(S116C2(:,i,j),3,3);
S117C2ij=reshape(S117C2(:,i,j),3,3);
S118C2ij=reshape(S118C2(:,i,j),3,3);

S124C2ij=reshape(S124C2(:,i,j),3,3);
S125C2ij=reshape(S125C2(:,i,j),3,3);
S126C2ij=reshape(S126C2(:,i,j),3,3);
S127C2ij=reshape(S127C2(:,i,j),3,3);

% C207 - WC207_06, WC207_15, WC207_24
errC207ij=errC207(i,j);

WC207_06ij=reshape(WC207_06(:,i,j),3,3);
newWC207_06ij=WC207_06ij+p*errC207ij*S106C2ij;
newWC207_06(:,i,j)=reshape(newWC207_06ij,1,[]);

WC207_15ij=reshape(WC207_15(:,i,j),3,3);
newWC207_15ij=WC207_15ij+p*errC207ij*S115C2ij;
newWC207_15(:,i,j)=reshape(newWC207_15ij,1,[]);

WC207_24ij=reshape(WC207_24(:,i,j),3,3);
newWC207_24ij=WC207_24ij+p*errC207ij*S124C2ij;
newWC207_24(:,i,j)=reshape(newWC207_24ij,1,[]);

% C208 - WC208_07, WC208_16, WC208_25
errC208ij=errC208(i,j);

WC208_07ij=reshape(WC208_07(:,i,j),3,3);
newWC208_07ij=WC208_07ij+p*errC208ij*S107C2ij;
newWC208_07(:,i,j)=reshape(newWC208_07ij,1,[]);

WC208_16ij=reshape(WC208_16(:,i,j),3,3);
newWC208_16ij=WC208_16ij+p*errC208ij*S116C2ij;
newWC208_16(:,i,j)=reshape(newWC208_16ij,1,[]);

WC208_25ij=reshape(WC208_25(:,i,j),3,3);
newWC208_25ij=WC208_25ij+p*errC208ij*S125C2ij;
newWC208_25(:,i,j)=reshape(newWC208_25ij,1,[]);

% C209 - WC209_08, WC209_17, WC209_26
errC209ij=errC209(i,j);

WC209_08ij=reshape(WC209_08(:,i,j),3,3);
newWC209_08ij=WC209_08ij+p*errC209ij*S108C2ij;
newWC209_08(:,i,j)=reshape(newWC209_08ij,1,[]);

WC209_17ij=reshape(WC209_17(:,i,j),3,3);
newWC209_17ij=WC209_17ij+p*errC209ij*S117C2ij;
newWC209_17(:,i,j)=reshape(newWC209_17ij,1,[]);

WC209_26ij=reshape(WC209_26(:,i,j),3,3);
newWC209_26ij=WC209_26ij+p*errC209ij*S126C2ij;

```

```

newWC209_26(:,i,j)=reshape(newWC209_26ij,1,[]);

% C210 - WC210_09, WC210_18, WC210_27
errC210ij=errC210(i,j);

WC210_09ij=reshape(WC210_09(:,i,j),3,3);
newWC210_09ij=WC210_09ij+p*errC210ij*S109C2ij;
newWC210_09(:,i,j)=reshape(newWC210_09ij,1,[]);

WC210_18ij=reshape(WC210_18(:,i,j),3,3);
newWC210_18ij=WC210_18ij+p*errC210ij*S118C2ij;
newWC210_18(:,i,j)=reshape(newWC210_18ij,1,[]);

WC210_27ij=reshape(WC210_27(:,i,j),3,3);
newWC210_27ij=WC210_27ij+p*errC210ij*S127C2ij;
newWC210_27(:,i,j)=reshape(newWC210_27ij,1,[]);

end
end

newbiasC201=biasC201+p*errC201;
newbiasC202=biasC202+p*errC202;
newbiasC203=biasC203+p*errC203;
newbiasC204=biasC204+p*errC204;
newbiasC205=biasC205+p*errC205;
newbiasC206=biasC206+p*errC206;
newbiasC207=biasC207+p*errC207;
newbiasC208=biasC208+p*errC208;
newbiasC209=biasC209+p*errC209;
newbiasC210=biasC210+p*errC210;

%% layer C1

for i=1:32
    for j=1:60
        % chanel R
        newWC101R(:,i,j)=WC101R(:,i,j)+p*errC101(i,j)*wR01(:,i,j);
        % chanel G
        newWC110G(:,i,j)=WC110G(:,i,j)+p*errC110(i,j)*wG01(:,i,j);
        %chanel B
        newWC119B(:,i,j)=WC119B(:,i,j)+p*errC119(i,j)*wB01(:,i,j);

    end
end

for i=1:14
    for j=1:28
        % chanel R
        newWC102R(:,i,j)=WC102R(:,i,j)+p*errC102(i,j)*wR02(:,i,j);
        newWC103R(:,i,j)=WC103R(:,i,j)+p*errC103(i,j)*wR03(:,i,j);
        newWC104R(:,i,j)=WC104R(:,i,j)+p*errC104(i,j)*wR04(:,i,j);
        newWC105R(:,i,j)=WC105R(:,i,j)+p*errC105(i,j)*wR05(:,i,j);
    end
end

```

```

% chanel G
newWC111G(:,i,j)=WC111G(:,i,j)+p*errC111(i,j)*wG11(:,i,j);
newWC112G(:,i,j)=WC112G(:,i,j)+p*errC112(i,j)*wG12(:,i,j);
newWC113G(:,i,j)=WC113G(:,i,j)+p*errC113(i,j)*wG13(:,i,j);
newWC114G(:,i,j)=WC114G(:,i,j)+p*errC114(i,j)*wG14(:,i,j);

%chanel B
newWC120B(:,i,j)=WC120B(:,i,j)+p*errC120(i,j)*wB20(:,i,j);
newWC121B(:,i,j)=WC121B(:,i,j)+p*errC121(i,j)*wB21(:,i,j);
newWC122B(:,i,j)=WC122B(:,i,j)+p*errC122(i,j)*wB22(:,i,j);
newWC123B(:,i,j)=WC123B(:,i,j)+p*errC123(i,j)*wB23(:,i,j);

end
end

for i=1:5
  for j=1:12

    % chanel R
    newWC106R(:,i,j)=WC106R(:,i,j)+p*errC106(i,j)*wR06(:,i,j);
    newWC107R(:,i,j)=WC107R(:,i,j)+p*errC107(i,j)*wR07(:,i,j);
    newWC108R(:,i,j)=WC108R(:,i,j)+p*errC108(i,j)*wR08(:,i,j);
    newWC109R(:,i,j)=WC109R(:,i,j)+p*errC109(i,j)*wR09(:,i,j);

    % chanel G
    newWC115G(:,i,j)=WC115G(:,i,j)+p*errC115(i,j)*wG15(:,i,j);
    newWC116G(:,i,j)=WC116G(:,i,j)+p*errC116(i,j)*wG16(:,i,j);
    newWC117G(:,i,j)=WC117G(:,i,j)+p*errC117(i,j)*wG17(:,i,j);
    newWC118G(:,i,j)=WC118G(:,i,j)+p*errC118(i,j)*wG18(:,i,j);

    %chanel B
    newWC124B(:,i,j)=WC124B(:,i,j)+p*errC124(i,j)*wB24(:,i,j);
    newWC125B(:,i,j)=WC125B(:,i,j)+p*errC125(i,j)*wB25(:,i,j);
    newWC126B(:,i,j)=WC126B(:,i,j)+p*errC126(i,j)*wB26(:,i,j);
    newWC127B(:,i,j)=WC127B(:,i,j)+p*errC127(i,j)*wB27(:,i,j);

  end
end

newbiasC101=biasC101+p      *errC109;      newbiasC118=biasC118+p
*errC101;                  newbiasC110=biasC110+p  *errC118;
newbiasC102=biasC102+p      *errC110;      newbiasC119=biasC119+p
*errC102;                  newbiasC111=biasC111+p  *errC119;
newbiasC103=biasC103+p      *errC111;      newbiasC120=biasC120+p
*errC103;                  newbiasC112=biasC112+p  *errC120;
newbiasC104=biasC104+p      *errC112;      newbiasC121=biasC121+p
*errC104;                  newbiasC113=biasC113+p  *errC121;
newbiasC105=biasC105+p      *errC113;      newbiasC122=biasC122+p
*errC105;                  newbiasC114=biasC114+p  *errC122;
newbiasC106=biasC106+p      *errC114;      newbiasC123=biasC123+p
*errC106;                  newbiasC115=biasC115+p  *errC123;
newbiasC107=biasC107+p      *errC115;      newbiasC124=biasC124+p
*errC107;                  newbiasC116=biasC116+p  *errC124;
newbiasC108=biasC108+p      *errC116;      newbiasC125=biasC125+p
*errC108;                  newbiasC117=biasC117+p  *errC125;
newbiasC109=biasC109+p      *errC117;      newbiasC126=biasC126+p

```



```

*errC126;                newbiasC127=biasC127+p        *errC127;

%% Save in cnet
% Layer C1

cnet.WC101R=WC101R;      cnet.WC110G=WC110G;      cnet.WC119B=WC119B;
cnet.WC102R=WC102R;      cnet.WC111G=WC111G;      cnet.WC120B=WC120B;
cnet.WC103R=WC103R;      cnet.WC112G=WC112G;      cnet.WC121B=WC121B;
cnet.WC104R=WC104R;      cnet.WC113G=WC113G;      cnet.WC122B=WC122B;
cnet.WC105R=WC105R;      cnet.WC114G=WC114G;      cnet.WC123B=WC123B;
cnet.WC106R=WC106R;      cnet.WC115G=WC115G;      cnet.WC124B=WC124B;
cnet.WC107R=WC107R;      cnet.WC116G=WC116G;      cnet.WC125B=WC125B;
cnet.WC108R=WC108R;      cnet.WC117G=WC117G;      cnet.WC126B=WC126B;
cnet.WC109R=WC109R;      cnet.WC118G=WC118G;      cnet.WC127B=WC127B;

cnet.biasC101=biasC101;  cnet.biasC110=biasC110;  cnet.biasC119=biasC119;
cnet.biasC102=biasC102;  cnet.biasC111=biasC111;  cnet.biasC120=biasC120;
cnet.biasC103=biasC103;  cnet.biasC112=biasC112;  cnet.biasC121=biasC121;
cnet.biasC104=biasC104;  cnet.biasC113=biasC113;  cnet.biasC122=biasC122;
cnet.biasC105=biasC105;  cnet.biasC114=biasC114;  cnet.biasC123=biasC123;
cnet.biasC106=biasC106;  cnet.biasC115=biasC115;  cnet.biasC124=biasC124;
cnet.biasC107=biasC107;  cnet.biasC116=biasC116;  cnet.biasC125=biasC125;
cnet.biasC108=biasC108;  cnet.biasC117=biasC117;  cnet.biasC126=biasC126;
cnet.biasC109=biasC109;  cnet.biasC118=biasC118;  cnet.biasC127=biasC127;

% Layer C2
cnet.WC201_01=newWC201_01; cnet.WC204_12=newWC204_12; cnet.WC207_24=newWC207_24;
cnet.WC201_10=newWC201_10; cnet.WC204_21=newWC204_21; cnet.WC208_07=newWC208_07;
cnet.WC201_19=newWC201_19; cnet.WC205_04=newWC205_04; cnet.WC208_16=newWC208_16;
cnet.WC202_01=newWC202_01; cnet.WC205_13=newWC205_13; cnet.WC208_25=newWC208_25;
cnet.WC202_10=newWC202_10; cnet.WC205_22=newWC205_22; cnet.WC209_08=newWC209_08;
cnet.WC202_19=newWC202_19; cnet.WC206_05=newWC206_05; cnet.WC209_17=newWC209_17;
cnet.WC203_02=newWC203_02; cnet.WC206_14=newWC206_14; cnet.WC209_26=newWC209_26;
cnet.WC203_11=newWC203_11; cnet.WC206_23=newWC206_23; cnet.WC210_09=newWC210_09;
cnet.WC203_20=newWC203_20; cnet.WC207_06=newWC207_06; cnet.WC210_18=newWC210_18;
cnet.WC204_03=newWC204_03; cnet.WC207_15=newWC207_15; cnet.WC210_27=newWC210_27;

```

```

cnet.biasC201=newbiasC201;
cnet.biasC202=newbiasC202;
cnet.biasC203=newbiasC203;
cnet.biasC204=newbiasC204;
cnet.biasC205=newbiasC205;
cnet.biasC206=newbiasC206;
cnet.biasC207=newbiasC207;
cnet.biasC208=newbiasC208;
cnet.biasC209=newbiasC209;
cnet.biasC210=newbiasC210;

```

```
% layer N1
```

```

cnet.WN101=newWN101;
cnet.WN102=newWN102;
cnet.WN103=newWN103;
cnet.WN104=newWN104;
cnet.WN105=newWN105;
cnet.WN106=newWN106;
cnet.WN107=newWN107;
cnet.WN108=newWN108;
cnet.WN109=newWN109;
cnet.WN110=newWN110;

```

```

cnet.biasN101=newbiasN101;
cnet.biasN102=newbiasN102;
cnet.biasN103=newbiasN103;
cnet.biasN104=newbiasN104;
cnet.biasN105=newbiasN105;
cnet.biasN106=newbiasN106;
cnet.biasN107=newbiasN107;
cnet.biasN108=newbiasN108;
cnet.biasN109=newbiasN109;
cnet.biasN110=newbiasN110;

```

```
% layerN2
```

```

cnet.WN2=newWN2;
cnet.biasN2=newbiasN2;

```

```
end
```

%Обучение свёрточных нейронных сетей для локализации текстовых %областей на полутоновых изображениях

```

close all;
clear all;
clc;

```

```

% load imdata_label_3730_29Sep_GrayImg;
% err_cnn=zeros(0);
% out_cnn=zeros(1);
% cnet=ConvNN_iBW();
%
% %
% circle=length(err_cnn);
% len_data=length(imdata);
% ep=0;
% i1=len_data*(ep-1)+1;
% i2=len_data*ep;
% max_err=2;

```

```
%% TRAIN
```

```

% load train_CNN_iBW_ex3730_ep16_rand_GrayImg.mat;
load train_CNN_iBW_ex2310_ep13_rand_GrayImg.mat;
len_data=length(imdata);
ep=13;
i1=len_data*(ep-1)+1;
i2=len_data*ep;
max_err=0;
for i=i1:i2
    if err_cnn(i)>max_err
        max_err=err_cnn(i);
    end
end

while (ep<15) && (max_err>0.01)
    ep=ep+1;
    p=0.1;
    len=length(imdata);
    for i=1:length(imdata)
        ir=round(1+(len-1).*rand(1,1));
        circle=circle+1;
        disp(circle);
        [cnet err_cnn out_cnn]= func_backpropagation_ConvNN_iBW_tangens_full
            (cnet,imdata,label,ir,p,err_cnn,out_cnn);
    end
    figure; plot(err_cnn(2:length(err_cnn)));
    i1=length(imdata)*(ep-1)+1;
    i2=length(imdata)*ep;
    max_err=0;
    for i=i1:i2
        if err_cnn(i)>max_err
            max_err=err_cnn(i);
        end
    end
end
end

```

%func_backpropagation_ConvNN_iBW_tangens_full.m

```

function [cnet err_cnn
out_cnn]=func_backpropagation_ConvNN_tangens_full(cnet,imdata,imlabel,ir,p,err_c
nn,out_cnn)

```

```

%% COEFFICIENT MATRIX (WEIGHT, BIAS)

```

```

% Layer C1

```

```

WC101=cnet.WC101;          biasC101=cnet.biasC101;
WC102=cnet.WC102;          biasC102=cnet.biasC102;
WC103=cnet.WC103;          biasC103=cnet.biasC103;
WC104=cnet.WC104;          biasC104=cnet.biasC104;
WC105=cnet.WC105;          biasC105=cnet.biasC105;
WC106=cnet.WC106;          biasC106=cnet.biasC106;
WC107=cnet.WC107;          biasC107=cnet.biasC107;
WC108=cnet.WC108;          biasC108=cnet.biasC108;
WC109=cnet.WC109;          biasC109=cnet.biasC109;

```

```

% Layer S1

```

```

WS1C101=cnet.WS1C101;
WS1C102=cnet.WS1C102;
WS1C103=cnet.WS1C103;
WS1C104=cnet.WS1C104;
WS1C105=cnet.WS1C105;
WS1C106=cnet.WS1C106;
WS1C107=cnet.WS1C107;
WS1C108=cnet.WS1C108;
WS1C109=cnet.WS1C109;

biasS101=cnet.biasS101;
biasS102=cnet.biasS102;
biasS103=cnet.biasS103;
biasS104=cnet.biasS104;
biasS105=cnet.biasS105;
biasS106=cnet.biasS106;
biasS107=cnet.biasS107;
biasS108=cnet.biasS108;
biasS109=cnet.biasS109;

% Layer C2

WC201_01=cnet.WC201_01;

WC202_02=cnet.WC202_02;
WC202_03=cnet.WC202_03;

WC203_04=cnet.WC203_04;
WC203_05=cnet.WC203_05;

WC204_06=cnet.WC204_06;
WC204_07=cnet.WC204_07;

WC205_08=cnet.WC205_08;
WC205_09=cnet.WC205_09;

biasC201=cnet.biasC201;
biasC202=cnet.biasC202;
biasC203=cnet.biasC203;
biasC204=cnet.biasC204;
biasC205=cnet.biasC205;

% layer S2
WS2C201=cnet.WS2C201;
WS2C202=cnet.WS2C202;
WS2C203=cnet.WS2C203;
WS2C204=cnet.WS2C204;
WS2C205=cnet.WS2C205;

% layer N1
WN101=cnet.WN101;
WN102=cnet.WN102;
WN103=cnet.WN103;
WN104=cnet.WN104;
WN105=cnet.WN105;

biasN101=cnet.biasN101;
biasN102=cnet.biasN102;
biasN103=cnet.biasN103;
biasN104=cnet.biasN104;
biasN105=cnet.biasN105;

% layerN2
WN2=cnet.WN2;
biasN2=cnet.biasN2;

%% DIRECT PASSAGE
label=imlabel(ir);
temp=imdata{ir};
iBW=temp;

%% CALCULATE ACTIVATOR OF LAYERS

%% Layer C1
[wBW, A1, H1, V1, D1, A2, H2, V2, D2] = func_directwavelet_level2( iBW );

```

```

for i=1:32
    for j=1:60
        % chanel R
        wBW01(:,i,j)=reshape(wBW(i:i+4,j:j+4),1,[]);
        sC101(i,j)=WC101(:,i,j)'*wBW01(:,i,j);

    end
end

for i=1:14
    for j=1:28

        wBW02(:,i,j)=reshape(A1(i:i+4,j:j+4),1,[]);
        sC102(i,j)=WC102(:,i,j)'*wBW02(:,i,j);

        wBW03(:,i,j)=reshape(H1(i:i+4,j:j+4),1,[]);
        sC103(i,j)=WC103(:,i,j)'*wBW03(:,i,j);

        wBW04(:,i,j)=reshape(V1(i:i+4,j:j+4),1,[]);
        sC104(i,j)=WC104(:,i,j)'*wBW04(:,i,j);

        wBW05(:,i,j)=reshape(D1(i:i+4,j:j+4),1,[]);
        sC105(i,j)=WC105(:,i,j)'*wBW05(:,i,j);

    end
end

for i=1:5
    for j=1:12

        wBW06(:,i,j)=reshape(A2(i:i+4,j:j+4),1,[]);
        sC106(i,j)=WC106(:,i,j)'*wBW06(:,i,j);

        wBW07(:,i,j)=reshape(H2(i:i+4,j:j+4),1,[]);
        sC107(i,j)=WC107(:,i,j)'*wBW07(:,i,j);

        wBW08(:,i,j)=reshape(V2(i:i+4,j:j+4),1,[]);
        sC108(i,j)=WC108(:,i,j)'*wBW08(:,i,j);

        wBW09(:,i,j)=reshape(D2(i:i+4,j:j+4),1,[]);
        sC109(i,j)=WC109(:,i,j)'*wBW09(:,i,j);

    end
end
end
sC101=sC101+biasC101;
sC102=sC102+biasC102;
sC103=sC103+biasC103;
sC104=sC104+biasC104;
sC105=sC105+biasC105;

```

```

sC106=sC106+biasC106;
sC107=sC107+biasC107;
sC108=sC108+biasC108;
sC109=sC109+biasC109;

uC101=(exp(sC101)-exp(-(sC101)))/(exp(sC101)+exp(-(sC101)));
uC102=(exp(sC102)-exp(-(sC102)))/(exp(sC102)+exp(-(sC102)));
uC103=(exp(sC103)-exp(-(sC103)))/(exp(sC103)+exp(-(sC103)));
uC104=(exp(sC104)-exp(-(sC104)))/(exp(sC104)+exp(-(sC104)));
uC105=(exp(sC105)-exp(-(sC105)))/(exp(sC105)+exp(-(sC105)));
uC106=(exp(sC106)-exp(-(sC106)))/(exp(sC106)+exp(-(sC106)));
uC107=(exp(sC107)-exp(-(sC107)))/(exp(sC107)+exp(-(sC107)));
uC108=(exp(sC108)-exp(-(sC108)))/(exp(sC108)+exp(-(sC108)));
uC109=(exp(sC109)-exp(-(sC109)))/(exp(sC109)+exp(-(sC109)));

%% Layer S1

for i=1:16
    i1=2*(i-1)+1;
    i2=i1+1;
    for j=1:30
        j1=2*(j-1)+1;
        j2=j1+1;
        uS101(i,j)=0.25*(sum(sum(uC101(i1:i2,j1:j2))));
    end
end
uS102=uC102;
uS103=uC103;
uS104=uC104;
uS105=uC105;
uS106=uC106;
uS107=uC107;
uS108=uC108;
uS109=uC109;

%% Layer C2

for i=1:14
    for j=1:28

        S101C2(:,i,j)=reshape(uS101(i:i+2,j:j+2),1,[]);
        S101C2ij=S101C2(:,i,j);
        %sC201
        WC201_1ij=WC201_01(:,i,j);
        sC201(i,j)=WC201_1ij'*S101C2ij;

    end
end

for i=1:12
    for j=1:26

        S102C2(:,i,j)=reshape(uS102(i:i+2,j:j+2),1,[]);
        S103C2(:,i,j)=reshape(uS103(i:i+2,j:j+2),1,[]);

```

```

S104C2(:,i,j)=reshape(uS104(i:i+2,j:j+2),1,[]);
S105C2(:,i,j)=reshape(uS105(i:i+2,j:j+2),1,[]);
%sC202
S102C2ij=S102C2(:,i,j);
S103C2ij=S103C2(:,i,j);

WC202_2ij=WC202_02(:,i,j);
WC202_3ij=WC202_03(:,i,j);
sC202(i,j)=WC202_2ij'*S102C2ij+WC202_3ij'*S103C2ij;

%sC203
S104C2ij=S104C2(:,i,j);
S105C2ij=S105C2(:,i,j);

WC203_4ij=WC203_04(:,i,j);
WC203_5ij=WC203_05(:,i,j);
sC203(i,j)=WC203_4ij'*S104C2ij+WC203_5ij'*S105C2ij;

end
end
for i=1:3
    for j=1:10

        S106C2(:,i,j)=reshape(uS106(i:i+2,j:j+2),1,[]);
        S107C2(:,i,j)=reshape(uS107(i:i+2,j:j+2),1,[]);
        S108C2(:,i,j)=reshape(uS108(i:i+2,j:j+2),1,[]);
        S109C2(:,i,j)=reshape(uS109(i:i+2,j:j+2),1,[]);

%sC204

S106C2ij=S106C2(:,i,j);
S107C2ij=S107C2(:,i,j);

WC204_6ij=WC204_06(:,i,j);
WC204_7ij=WC204_07(:,i,j);

sC204(i,j)=WC204_6ij'*S106C2ij+WC204_7ij'*S107C2ij;

%sC205
S108C2ij=S108C2(:,i,j);
S109C2ij=S109C2(:,i,j);

WC205_8ij=WC205_08(:,i,j);
WC205_9ij=WC205_09(:,i,j);

sC205(i,j)=WC205_8ij'*S108C2ij+WC205_9ij'*S109C2ij;

end
end

%uC201
sC201=sC201+biasC201;
uC201=(exp(sC201)-exp(-(sC201)))/(exp(sC201)+exp(-(sC201)));
%uC202

```

```

sC202=sC202+biasC202;
uC202=(exp(sC202)-exp(-(sC202)))/(exp(sC202)+exp(-(sC202)));
%uC203
sC203=sC203+biasC203;
uC203=(exp(sC203)-exp(-(sC203)))/(exp(sC203)+exp(-(sC203)));
%uC204
sC204=sC204+biasC204;
uC204=(exp(sC204)-exp(-(sC204)))/(exp(sC204)+exp(-(sC204)));
%uC205
sC205=sC205+biasC205;
uC205=(exp(sC205)-exp(-(sC205)))/(exp(sC205)+exp(-(sC205)));

%% layer S2
for i=1:7
    i1=2*(i-1)+1;
    i2=i1+1;
    for j=1:14
        j1=2*(j-1)+1;
        j2=j1+1;
        uS201(i,j)=0.25*(sum(sum(uC201(i1:i2,j1:j2))));
    end
end
for i=1:6
    i1=2*(i-1)+1;
    i2=i1+1;
    for j=1:13
        j1=2*(j-1)+1;
        j2=j1+1;
        uS202(i,j)=0.25*(sum(sum(uC202(i1:i2,j1:j2))));
        uS203(i,j)=0.25*(sum(sum(uC203(i1:i2,j1:j2))));
    end
end
uS204=uC204;
uS205=uC205;

%% layer N1
sN101=sum(sum(WN101.*uS201))/98+biasN101;
sN102=sum(sum(WN102.*uS202))/98+biasN102;
sN103=sum(sum(WN103.*uS203))/98+biasN103;
sN104=sum(sum(WN104.*uS204))/98+biasN104;
sN105=sum(sum(WN105.*uS205))/98+biasN105;

uN101=(exp(sN101)-exp(-(sN101)))/(exp(sN101)+exp(-(sN101)));
uN102=(exp(sN102)-exp(-(sN102)))/(exp(sN102)+exp(-(sN102)));
uN103=(exp(sN103)-exp(-(sN103)))/(exp(sN103)+exp(-(sN103)));
uN104=(exp(sN104)-exp(-(sN104)))/(exp(sN104)+exp(-(sN104)));
uN105=(exp(sN105)-exp(-(sN105)))/(exp(sN105)+exp(-(sN105)));

%% layer N2
tam1=WN2(1,1)*uN101+WN2(1,2)*uN102+WN2(1,3)*uN103+WN2(1,4)*uN104+WN2(1,5)*uN105;
sN2=tam1+biasN2;
uN2=(exp(sN2)-exp(-sN2))/(exp(sN2)+exp(-sN2));
out=uN2;
out_cnn=[out_cnn out];

```



```

%% CALCULATE ERRORS OF LAYERS
C=label;
e=0.5*((C-out)^2);
err_cnn=[err_cnn e];
%% layer N2
errN2=(C-out)*(1-out)*(1+out)/2;

%% layer N1
errN101=0.5*(1+uN101)*(1-uN101)*(errN2*WN2(1,1));
errN102=0.5*(1+uN102)*(1-uN102)*(errN2*WN2(1,2));
errN103=0.5*(1+uN103)*(1-uN103)*(errN2*WN2(1,3));
errN104=0.5*(1+uN104)*(1-uN104)*(errN2*WN2(1,4));
errN105=0.5*(1+uN105)*(1-uN105)*(errN2*WN2(1,5));

%% layer S2

errS201=0.5*(1+uS201).*(1-uS201).*(errN101*WN101);
errS202=0.5*(1+uS202).*(1-uS202).*(errN102*WN102);
errS203=0.5*(1+uS203).*(1-uS203).*(errN103*WN103);
errS204=0.5*(1+uS204).*(1-uS204).*(errN104*WN104);
errS205=0.5*(1+uS205).*(1-uS205).*(errN105*WN105);

%% layer C2
for i=1:7
    i1=2*(i-1)+1;
    i2=i1+1;
    for j=1:14
        j1=2*(j-1)+1;
        j2=j1+1;
        errC201(i1:i2,j1:j2)=errS201(i,j).*WS2C201(i,j);
    end
end
for i=1:6
    i1=2*(i-1)+1;
    i2=i1+1;
    for j=1:13
        j1=2*(j-1)+1;
        j2=j1+1;
        errC202(i1:i2,j1:j2)=errS202(i,j).*WS2C202(i,j);
        errC203(i1:i2,j1:j2)=errS203(i,j).*WS2C203(i,j);
    end
end
for i=1:3
    for j=1:10
        errC204(i,j)=errS204(i,j).*WS2C204(i,j);
        errC205(i,j)=errS205(i,j).*WS2C205(i,j);
    end
end

%% Layer S1
rWC201_01=zeros(16,30);
for i=1:14
    for j=1:28
        w=errC201(i,j)*WC201_01(:,i,j);
        ww=reshape(w,3,3);
    end
end

```

```

        rWC201_01(i:i+2,j:j+2)=rWC201_01(i:i+2,j:j+2)+ww;
    end
end
errS101=0.5*(1+uS101).*(1-uS101).*rWC201_01;

rWC202_02=zeros(14,28);
rWC202_03=zeros(14,28);

rWC203_04=zeros(14,28);
rWC203_05=zeros(14,28);

for i=1:12
    ii=i;
    for j=1:26
        jj=j;

        w=errC202(i,j)*WC202_02(:,i,j);
        ww=reshape(w,3,3);
        rWC202_02(ii:ii+2,jj:jj+2)=rWC202_02(ii:ii+2,jj:jj+2)+ww;

        w=errC202(i,j)*WC202_03(:,i,j);
        ww=reshape(w,3,3);
        rWC202_03(ii:ii+2,jj:jj+2)=rWC202_03(ii:ii+2,jj:jj+2)+ww;

        w=errC203(i,j)*WC203_04(:,i,j);
        ww=reshape(w,3,3);
        rWC203_04(ii:ii+2,jj:jj+2)=rWC203_04(ii:ii+2,jj:jj+2)+ww;

        w=errC203(i,j)*WC203_05(:,i,j);
        ww=reshape(w,3,3);
        rWC203_05(ii:ii+2,jj:jj+2)=rWC203_05(ii:ii+2,jj:jj+2)+ww;

    end
end
errS102=0.5*(1+uS102).*(1-uS102).*rWC202_02;
errS103=0.5*(1+uS103).*(1-uS103).*rWC202_03;
errS104=0.5*(1+uS104).*(1-uS104).*rWC203_04;
errS105=0.5*(1+uS105).*(1-uS105).*rWC203_05;

rWC204_06=zeros(5,12);
rWC204_07=zeros(5,12);

rWC205_08=zeros(5,12);
rWC205_09=zeros(5,12);

for i=1:3
    ii=i;
    for j=1:10
        jj=j;

        w=errC204(i,j)*WC204_06(:,i,j);
        ww=reshape(w,3,3);
        rWC204_06(ii:ii+2,jj:jj+2)=rWC204_06(ii:ii+2,jj:jj+2)+ww;
    end
end

```

```

w=errC204(i,j)*WC204_07(:,i,j);
ww=reshape(w,3,3);
rWC204_07(ii:ii+2,jj:jj+2)=rWC204_07(ii:ii+2,jj:jj+2)+ww;

w=errC205(i,j)*WC205_08(:,i,j);
ww=reshape(w,3,3);
rWC205_08(ii:ii+2,jj:jj+2)=rWC205_08(ii:ii+2,jj:jj+2)+ww;

w=errC205(i,j)*WC205_09(:,i,j);
ww=reshape(w,3,3);
rWC205_09(ii:ii+2,jj:jj+2)=rWC205_09(ii:ii+2,jj:jj+2)+ww;

    end
end

errS106=0.5*(1+uS106).*(1-uS106).*rWC204_06;
errS107=0.5*(1+uS107).*(1-uS107).*rWC204_07;
errS108=0.5*(1+uS108).*(1-uS108).*rWC205_08;
errS109=0.5*(1+uS109).*(1-uS109).*rWC205_09;

%% layer C1
for i=1:16
    i1=2*(i-1)+1;
    i2=i1+1;
    for j=1:30
        j1=2*(j-1)+1;
        j2=j1+1;
        errC101(i1:i2,j1:j2)=errS101(i,j).*WS1C101(i,j);
    end
end
errC102=errS102;
errC103=errS103;
errC104=errS104;
errC105=errS105;
errC106=errS106;
errC107=errS107;
errC108=errS108;
errC109=errS109;

%% CHANGE WEIGHT AND BIAS
% layer N2
newWN2(1,1)=WN2(1,1)+p*errN2*uN101;
newWN2(1,2)=WN2(1,2)+p*errN2*uN102;
newWN2(1,3)=WN2(1,3)+p*errN2*uN103;
newWN2(1,4)=WN2(1,4)+p*errN2*uN104;
newWN2(1,5)=WN2(1,5)+p*errN2*uN105;

newbiasN2=biasN2+p*errN2;

%% layer N1
newWN101=WN101+p*errN101*uS201;
newWN102=WN102+p*errN102*uS202;
newWN103=WN103+p*errN103*uS203;
newWN104=WN104+p*errN104*uS204;
newWN105=WN105+p*errN105*uS205;
newbiasN101=biasN101+p*errN101;
newbiasN102=biasN102+p*errN102;
newbiasN103=biasN103+p*errN103;
newbiasN104=biasN104+p*errN104;

```

```

newbiasN105=biasN105+p*errN105;

%% layer C2

for i=1:14
    for j=1:28

        S101C2ij=reshape(S101C2(:,i,j),3,3);

% C201 - WC201_01
        errC201ij=errC201(i,j);

        WC201_01ij=reshape(WC201_01(:,i,j),3,3);
        newWC201_01ij=WC201_01ij+p*errC201ij*S101C2ij;
        newWC201_01(:,i,j)=reshape(newWC201_01ij,1,[]);

    end
end

for i=1:12
    for j=1:26

        S102C2ij=reshape(S102C2(:,i,j),3,3);
        S103C2ij=reshape(S103C2(:,i,j),3,3);
        S104C2ij=reshape(S104C2(:,i,j),3,3);
        S105C2ij=reshape(S105C2(:,i,j),3,3);

% C202 - WC202_02, WC202_03
        errC202ij=errC202(i,j);

        WC202_02ij=reshape(WC202_02(:,i,j),3,3);
        newWC202_02ij=WC202_02ij+p*errC202ij*S102C2ij;
        newWC202_02(:,i,j)=reshape(newWC202_02ij,1,[]);

        WC202_03ij=reshape(WC202_03(:,i,j),3,3);
        newWC202_03ij=WC202_03ij+p*errC202ij*S103C2ij;
        newWC202_03(:,i,j)=reshape(newWC202_03ij,1,[]);

% C203 - WC203_04, WC203_05
        errC203ij=errC203(i,j);

        WC203_04ij=reshape(WC203_04(:,i,j),3,3);
        newWC203_04ij=WC203_04ij+p*errC203ij*S104C2ij;
        newWC203_04(:,i,j)=reshape(newWC203_04ij,1,[]);

        WC203_05ij=reshape(WC203_05(:,i,j),3,3);
        newWC203_05ij=WC203_05ij+p*errC203ij*S105C2ij;
        newWC203_05(:,i,j)=reshape(newWC203_05ij,1,[]);

    end
end
end

```

```

for i=1:3
    for j=1:10

        S106C2ij=reshape(S106C2(:,i,j),3,3);
        S107C2ij=reshape(S107C2(:,i,j),3,3);
        S108C2ij=reshape(S108C2(:,i,j),3,3);
        S109C2ij=reshape(S109C2(:,i,j),3,3);

        % C204 - WC204_06, WC204_07
        errC204ij=errC204(i,j);

        WC204_06ij=reshape(WC204_06(:,i,j),3,3);
        newWC204_06ij=WC204_06ij+p*errC204ij*S106C2ij;
        newWC204_06(:,i,j)=reshape(newWC204_06ij,1,[]);

        WC204_07ij=reshape(WC204_07(:,i,j),3,3);
        newWC204_07ij=WC204_07ij+p*errC204ij*S107C2ij;
        newWC204_07(:,i,j)=reshape(newWC204_07ij,1,[]);

        % C205 - WC205_08, WC205_09
        errC205ij=errC205(i,j);

        WC205_08ij=reshape(WC205_08(:,i,j),3,3);
        newWC205_08ij=WC205_08ij+p*errC205ij*S108C2ij;
        newWC205_08(:,i,j)=reshape(newWC205_08ij,1,[]);

        WC205_09ij=reshape(WC205_09(:,i,j),3,3);
        newWC205_09ij=WC205_09ij+p*errC205ij*S109C2ij;
        newWC205_09(:,i,j)=reshape(newWC205_09ij,1,[]);

    end
end

newbiasC201=biasC201+p*errC201;
newbiasC202=biasC202+p*errC202;
newbiasC203=biasC203+p*errC203;
newbiasC204=biasC204+p*errC204;
newbiasC205=biasC205+p*errC205;

%% layer C1

for i=1:32
    for j=1:60
        newWC101(:,i,j)=WC101(:,i,j)+p*errC101(i,j)*wBW01(:,i,j);
    end
end

for i=1:14
    for j=1:28
        newWC102(:,i,j)=WC102(:,i,j)+p*errC102(i,j)*wBW02(:,i,j);
        newWC103(:,i,j)=WC103(:,i,j)+p*errC103(i,j)*wBW03(:,i,j);
        newWC104(:,i,j)=WC104(:,i,j)+p*errC104(i,j)*wBW04(:,i,j);
        newWC105(:,i,j)=WC105(:,i,j)+p*errC105(i,j)*wBW05(:,i,j);
    end
end

```

```

end

for i=1:5
    for j=1:12
        newWC106(:,i,j)=WC106(:,i,j)+p*errC106(i,j)*wBW06(:,i,j);
        newWC107(:,i,j)=WC107(:,i,j)+p*errC107(i,j)*wBW07(:,i,j);
        newWC108(:,i,j)=WC108(:,i,j)+p*errC108(i,j)*wBW08(:,i,j);
        newWC109(:,i,j)=WC109(:,i,j)+p*errC109(i,j)*wBW09(:,i,j);
    end
end

newbiasC101=biasC101+p*errC101;
newbiasC102=biasC102+p*errC102;
newbiasC103=biasC103+p*errC103;
newbiasC104=biasC104+p*errC104;
newbiasC105=biasC105+p*errC105;
newbiasC106=biasC106+p*errC106;
newbiasC107=biasC107+p*errC107;
newbiasC108=biasC108+p*errC108;
newbiasC109=biasC109+p*errC109;

%% Save in cnet
% Layer C1
cnet.WC101=newWC101;
cnet.WC102=newWC102;
cnet.WC103=newWC103;
cnet.WC104=newWC104;
cnet.WC105=newWC105;
cnet.WC106=newWC106;
cnet.WC107=newWC107;
cnet.WC108=newWC108;
cnet.WC109=newWC109;

cnet.biasC101=biasC101;
cnet.biasC102=biasC102;
cnet.biasC103=biasC103;
cnet.biasC104=biasC104;
cnet.biasC105=biasC105;
cnet.biasC106=biasC106;
cnet.biasC107=biasC107;
cnet.biasC108=biasC108;
cnet.biasC109=biasC109;

% Layer C2
cnet.WC201_01=newWC201_01;

cnet.WC202_02=newWC202_02;
cnet.WC202_03=newWC202_03;

cnet.WC203_04=newWC203_04;
cnet.WC203_05=newWC203_05;

cnet.WC204_06=newWC204_06;
cnet.WC204_07=newWC204_07;

cnet.WC205_08=newWC205_08;
cnet.WC205_09=newWC205_09;

cnet.biasC201=newbiasC201;
cnet.biasC202=newbiasC202;
cnet.biasC203=newbiasC203;
cnet.biasC204=newbiasC204;
cnet.biasC205=new
biasC205;

% layer N1
cnet.WN101=newWN101;
cnet.WN102=newWN102;
cnet.WN103=newWN103;
cnet.WN104=newWN104;
cnet.WN105=newWN105;

cnet.biasN102=newbiasN102;
cnet.biasN103=newbiasN103;
cnet.biasN104=newbiasN104;
cnet.biasN105=newbiasN105;

cnet.biasN101=newbiasN101;

```

```
% layerN2
cnet.WN2=newWN2;
cnet.biasN2=newbiasN2;
```

```
end
```

```
%Программы тестирования локализации ТО на изображениях
```

```
  % func_simplify_detect.m
  % sim_ConvNN_test.m
  % sim_ConvNN_iBW_test.m
  % func_imgmov50_process_sos.m
  % func_directwavelet_level2.m
  % func_improved_contrast_RGB_v4.m
  % func_preprocess_image.m
  % func_segment_rows.m
  % func_define_located_TO.m
  % func_define_subIm_input.m
```

```
%TEST_ConvNN_testing_REALimages_7August.m
```

```
close all;
clear all;
clc;
%% ***** LOAD CNN *****
load train_CNN_ex3730_ep41_rand_7August.mat;

%% ***** TEST WITH REAL IMAGE *****
% path = 'English Images\104.jpg';
path = 'ICDAR2011\dataset_images\img_396.png';
RGB = imread(path);
[row col]=size(RGB);
const=36*10;
if row>const
    nr=const;
    scale=nr/row;
    img_input=imresize(RGB,scale,'bilinear');
else
    img_input=RGB;
end
figure, imshow(img_input);
%% SIMPLY DETECT CADIDATE TEXT REGION IN IMAGE
% result - subIm1 --> the second stage - locate using CNN
% result - subIm2 --> segmentation
temp1 = img_input;
temp2=double(temp1)/256;
[row col]=size(temp2);
img_input=temp2;
figure, imshow(img_input);
[ subIm, coord_subIm, subIm1, subIm2 ] = func_simplify_detect( img_input );
for k=1:length(subIm)
% PRE-PROCESSING IMAGE
    window=[18 32;36 64;54 96;72 128;90 160;108 192;126 224];
    num=2; %num=1,...,7
    wwind =window(num,1);
    hwind=window(num,2);
    image=subIm{k};
```

```

[ir ic]=size(image);
dir=ir/36;
dic=ic/64;
if (dir<1) || (dic<1)
    if dir<=dic
        scale=ceil(36/ir);
    else
        scale=ceil(64/ic);
    end
    nim=imresize(image,scale,'bilinear');
    image=nim;
end
% figure,imshow(image);
[new_img imgs sub isquant iscoord]=func_imgmov50_process_sos(image,
wwind,hwind);

%% USING CNN
tic();
result=zeros(isquant,2);
for i=1:isquant
    disp(i);
    result(i,1)=i;
    out=sim_ConvNN_test(cnet,imgs sub,i);
    result(i,2)=out;
end;
res_coord= iscoord;
[nr nc n]=size(new_img);
res_img=zeros(nr, nc, n);
res_img_better=zeros(nr, nc, n);
res=round(result(:,2));
for i=1:isquant
    if res(i)==1
        i1=res_coord(i,1);
        i2=res_coord(i,2);
        j1=res_coord(i,3);
        j2=res_coord(i,4);
        res_img(i1:i2,j1:j2,:)=new_img(i1:i2,j1:j2,:);
    end
    if res(i)==1
        i1=res_coord(i,1);
        i2=res_coord(i,2);
        j1=res_coord(i,3);
        j2=res_coord(i,4);
        if i1>5
            i1=i1-5;
        end
        if i2<(size(new_img,1)-5)
            i2=i2+5;
        end
        if j1>20
            j1=j1-20;
        end
        if j2<(size(new_img,2)-15)
            j2=j2+15;
        end
        res_img_better(i1:i2,j1:j2,:)=new_img(i1:i2,j1:j2,:);
    end
end
sub_resRGB{k}=res_img;
sub_resRGB_better{k}=res_img_better;
end

%% FIND BOUNDINGBOX AND DRAW RECTANGLE

```



```

for k=1:length(subIm)
    [row1 col1 n1]=size(subIm{k});
    newsub_resRGB{k}=imresize(sub_resRGB{k},[row1 col1],'bilinear');
    newsub_resRGB_better{k}=imresize(sub_resRGB_better{k},[row1
coll1],'bilinear');
end
[row col n]=size(img_input);
img_text=img_input;
img_text=zeros(row,col,n);
for k=1:length(subIm)
    i1=coord_subIm(k,1);
    i2=coord_subIm(k,2);
    j1=coord_subIm(k,3);
    j2=coord_subIm(k,4);
    img_text(i1:i2,j1:j2,:)=newsub_resRGB{k};
    img_text_better(i1:i2,j1:j2,:)=newsub_resRGB_better{k};
end
figure,imshow(img_text);
figure,imshow(img_text_better);

```

% func_simple_detect.m

```

function [ subIm, coord_subIm, subIm1, subIm2 ] = func_simple_detect(
img_input )

%% SIMPLY DETECT CADIDATE TEXT REGION IN IMAGE
% result - subIm1 --> using CNN
% result - subIm2 --> segmentation
I=img_input;
Igray=rgb2gray(I);
Ibw = edge(Igray,'canny');

% Dilation image
nr1=5;
nc1=21;
cross1=zeros(nr1,nc1);
n1=(nr1-1)/2+1;
n2=(nc1-1)/2+1;
cross1(n1,:)=1;
cross1(:,n2)=1;
SE1= strel('arbitrary',cross1);
BW1 = imdilate(Ibw, SE1);
% figure(), imshow(BW1);

% Close image
nr2=11;
nc2=45;
cross2=zeros(nr2,nc2);
n1=(nr2-1)/2+1;
n2=(nc2-1)/2+1;
cross2(n1,:)=1;
cross2(:,n2)=1;
SE2 = strel('arbitrary',cross2);
BW2 = imopen(BW1, SE2);
% figure(), imshow(BW2);

% Locate cadidated text region
[rowI colI]=size(BW2);
threshX=colI/3;
threshY=rowI/3;

```

```

stats = regionprops (BW2);
num1=0;
num2=0;
num=0;
res_local=img_input;
for index=1:length(stats)
    if stats(index).Area > 200 &&
stats(index).BoundingBox(3)*stats(index).BoundingBox(4) < 500000
        x = ceil(stats(index).BoundingBox(1));
        y= ceil(stats(index).BoundingBox(2));
        widthX = floor(stats(index).BoundingBox(3)-1);
        widthY = floor(stats(index).BoundingBox(4)-1);
        x1=x;
        y1=y;
        x2=x1+widthX;
        y2=y1+widthY;
        tr=1;
        if (widthX>=threshX) && (widthY>=threshY)
            num1=num1+1;
            coord_BW1(num1,:)=[y y+widthY x x+widthX];
            subBW1{num1}=I(y:y+widthY, x:x+widthX);
            subIm1{num1}=I(y:y+widthY, x:x+widthX,:);
            num=num+1;
            subIm{num}=I(y:y+widthY, x:x+widthX,:);
            coord_subIm(num,:)=[y y+widthY x x+widthX];
            % subIm1 --> using CNN
            res_local(y:y+2, x:x+widthX,:)=tr;
            res_local(y+widthY-2:y+widthY, x:x+widthX,:)=tr;
            res_local(y:y+widthY,x:x+2)=tr;
            res_local(y:y+widthY,x+widthX-2:x+widthX)=tr;

        else
            num2=num2+1;
            coord_BW2(num2,:)=[y y+widthY x x+widthX];
            subBW2{num2}=I(y:y+widthY, x:x+widthX);
            subIm2{num2}.image=I(y:y+widthY, x:x+widthX,:);
            subIm2{num2}.coord=[y y+widthY x x+widthX];
            num=num+1;
            subIm{num}=I(y:y+widthY, x:x+widthX,:);
            coord_subIm(num,:)=[y y+widthY x x+widthX];
            % subIm2 --> segmentation
            res_local(y:y+2, x:x+widthX,:)=tr;
            res_local(y+widthY-2:y+widthY, x:x+widthX,:)=tr;
            res_local(y:y+widthY,x:x+2)=tr;
            res_local(y:y+widthY,x+widthX-2:x+widthX)=tr;
        end
    end
end
if num1==0
    subIm1{1}=zeros(1);
end
if num2==0
    subIm2{1}=zeros(1);
end
if num==0
    subIm{1}=zeros(1);
    coord_subIm=zeros(1,4);
end
figure,imshow(res_local);title('result_location_step');
end

```

% func_imgmov50_process_sos.m

```

function [new_img imgsub isquant iscoord]=func_imgmov50_process_sos(img,
wwind,hwind)
[nrow1 ncol1 nrc]=size(img);
inr=round(nrow1/wwind);
inc=round(ncol1/hwind);
nrow=inr*wwind;
ncol=inc*hwind;
new_img=imresize(img,[nrow ncol],'bilinear');
num=0;
for i=1:wwind/2:nrow-wwind+1
    for j=1:hwind/2:ncol-hwind+1
        num=num+1;
    end
end
isquant=0;
iscoord=zeros(num,4);
for i=1:wwind/2:nrow-wwind+1
    i1=i;
    i2=i+wwind-1;
    for j=1:hwind/2:ncol-hwind+1
        j1=j;
        j2=j+hwind-1;
        isquant=isquant+1;
        ij=[i1 i2 j1 j2];
        imgsub{isquant}=imresize(new_img(i1:i2,j1:j2,:),[36 64],'bilinear');
        iscoord(isquant,:)=ij;
    end
end
end

function [wR, RA1, RH1, RV1, RD1, RA2, RH2, RV2, RD2] =
func_directwavelet_level2( R )

[RA1, RH1, RV1, RD1] = dwt2(R,'db1');
[RA2, RH2, RV2, RD2] = dwt2(RA1,'db1');
wR=R;

end

```

% func_directwavelet_level2.m

```

function [wR, RA1, RH1, RV1, RD1, RA2, RH2, RV2, RD2] =
func_directwavelet_level2( R )

[RA1, RH1, RV1, RD1] = dwt2(R,'db1');
[RA2, RH2, RV2, RD2] = dwt2(RA1,'db1');
wR=R;

end

```

% func_improved_contrast_RGB_v4.m

```

function [ new_img ] = func_improved_contrast_RGB_v4( image )

RGB = image;
for ch=1:3
    RGB(:, :, ch) = imadjust( RGB(:, :, ch), stretchlim( RGB(:, :, ch), 0.01) );
end
img_adjusted = rgb2gray( RGB );
new_img=img_adjusted;

```

```
end
```

% func_preprocess_image.m

```
function [ output_image ] = func_preprocess_image( input_image )
```

```
%% preprocess_image before segmentation stage
```

```
z1=input_image;
z2=double(z1);
z3=z2.^3;
z4=z3./(256*2);
t=(z4-min(min(z4)));
tt=(max(max(z4))-min(min(z4)));
z5=(t./tt)*255;
z6=uint8(z5);
z7=im2double(z6);
[n xout]=hist(reshape(z7,1,[]));
maxn=max(n);
for kk=1:length(n)
    if n(kk)==maxn
        pp=kk;
    end
end
if (xout(pp)<0.5)
    z8=1-z7;
else
    z8=z7;
end
output_image =z8;
% figure, imshow(z9);
```

```
end
```

% func_segment_rows.m

```
function [ segrows ] = func_segment_rows( input_image, coord_input )
```

```
% segment rows of image
```

```
jbegin=coord_input(1,3);
jend=coord_input(1,4);
z8=input_image;
zim=1-z8;
[nrow ncol]=size(zim);
sj=zeros(1);
for j=1:nrow
    sj(j)=sum(zim(j,:))/ncol;
end
s=sum(sj)/nrow;
kt=0.5;
st=kt*s;

nib=0;
ibegin=zeros(1);
for i=1:length(sj)-1
    if (sj(i)<=st) && (sj(i+1)>st)
        nib=nib+1;
        ibegin(nib)=i+1;
    end
    if (i==1) && (sj(i)>st)
        nib=nib+1;
        ibegin(nib)=i;
    end
end
```

```

end
nie=0;
iend=zeros(1);

for k=1:nib
    i1=ibegin(k);
    if (k+1)>nib
        i2=length(sj);
    else
        i2=ibegin(k+1)-1;
    end
    for i=i1:i2-1
        if ((sj(i)>st) && (sj(i+1)<st))
            nie=nie+1;
            iend(nie)=i;
        end
        if (i==i2-1) && (sj(i+1)>st) && (sj(i)>st)
            nie=nie+1;
            iend(nie)=i+1;
        end
    end
end
end
%
ii=0;
segrows{1}.img=zeros(1);
segrows{1}.icoord=zeros(1);
for i=1:nib
    if (iend(i)-ibegin(i)>5)
        ii=ii+1;
        Hmin=(iend(i)-ibegin(i))+1;
        Hadd=round(0.3*Hmin);
        H=Hmin+2*Hadd;
        if ibegin(i)>Hadd
            ibegin1(i)=ibegin(i)-Hadd;
        else
            ibegin1(i)=1;
        end
        if (length(sj)-iend(i))>Hadd
            iend1(i)=iend(i)+Hadd;
        else
            iend1(i)=length(sj);
        end
        zim1=zim(ibegin1(i):iend1(i),:);
        % figure,imshow(zim1);
        ss1=sum(zim1,1);
        zbegin=1;
        iz=1;
        test=0;
        while (iz<length(ss1)-1) && (test==0)
            if (ss1(iz)>0) || ((ss1(iz)==0) && ss1(iz+1)>0)
                zbegin=iz;
                test=1;
            end
            if (ss1(iz)==0) && ss1(iz+1)>0
                zbegin=iz+1;
                test=1;
            end
            iz=iz+1;
        end
        zend=length(ss1);
        iz=length(ss1);
        test=0;
        while (iz>2) && (test==0)

```

```

        if (ss1(iz)>0)
            zend=iz;
            test=1;
        end
        if ((ss1(iz)==0) && ss1(iz-1)>0)
            zend=iz-1;
            test=1;
        end
        iz=iz-1;
    end
    zim2=zim1(:,zbegin:zend);
    segrows{ii}.img=zim2;
%     segrows{ii}.icoord=[ibeginl(i)+coord_input(1,1)-1
iendl(i)+coord_input(1,1)-1 jbegin jend];
    segrows{ii}.icoord=[ibeginl(i)+coord_input(1,1)-1
iendl(i)+coord_input(1,1)-1 zbegin+jbegin-1 zend+jbegin-1];
    end
end
end
end

```

% func_define_located_TO.m

```

function [ locatedTO ] = func_define_located_TO( subIm_input )

knum=length(subIm_input);
num_textlocated=0;
for k=1:knum

    img=subIm_input{k}.img;
    coord_input=subIm_input{k}.coord;
%     figure, imshow(img);

    %% Preprocess image
    z8= func_preprocess_image(img);
    % z9=2*z8-1;
    z8=imadjust(z8,[0.15 mean(mean(z8))],[,]);
    %% segment rows
    input_image=z8;
    [segrows] = func_segment_rows(input_image, coord_input);
    text=1;
    if segrows{1}.icoord(1)>0
        nnew=0;
        for i=1:length(segrows)
            sr=segrows{i}.img;
            ss1=sum(sr,1);
            % j begin
            jb_sr=zeros(0);
            njb=0;
            if (ss1(1)>0)
                njb=njb+1;
                jb_sr(njb)=1;
            end
            for j=1:length(ss1)-3
                if (ss1(j)==0) && (ss1(j+1)==0) && (ss1(j+2)==0) &&
(ss1(j+3)>0)
                    njb=njb+1;
                    jb_sr(njb)=j+3;
                end
            end
        end
        % j end
        je_sr=zeros();
    end
end

```

```

        nje=0;
        for j=1:length(ss1)-3
            if (ss1(j)>0) && (ss1(j+1)==0) && (ss1(j+2)==0) &&
(ss1(j+3)==0)
                nje=nje+1;
                je_sr(nje)=j;
            end
        end
        if (ss1(length(ss1))>0)
            nje=nje+1;
            je_sr(nje)=length(ss1);
        end
        % new segrows
        iii=nnew;
        for ii=nnew+1:nnew+njb
            i1=segrows{i}.icoord(1);
            i2=segrows{i}.icoord(2);
            j1=jb_sr(ii-nnew);
            j2=je_sr(ii-nnew);
            temp=im2bw(sr(:,j1:j2));
            sss=sum(sum(temp));
            if sss>25
                iii=iii+1;
                new_sr=sr(:,j1:j2);
                new_segrows{iii}.img=new_sr;
                new_segrows{iii}.coord=[i1 i2 j1+segrows{i}.icoord(3)-1
j2+segrows{i}.icoord(3)-1];
                % figure,imshow(new_sr);
                % disp([i1 i2 j1 j2]);
            end

        end
        nnew=iii;
    end
    nsr=length(new_segrows);
    for nt=num_textlocated+1:num_textlocated+nsr
        nti=nt-num_textlocated;
        locatedTO{nt}.img= new_segrows{nti}.img;
        locatedTO{nt}.coord= new_segrows{nti}.coord;
    end
    num_textlocated=num_textlocated+nnew;
    clear new_segrows;
else
    locatedTO{nt}.img= zeros(0);
    locatedTO{nt}.coord=zeros(0);
end
end
end
end

```

% func_define_subIm_input.m

```

function [ subIm_input ] = func_define_subIm_input( input )

% define coordinates TO
s1=sum(input,2);
ibegin=zeros(0);
nib=0;
if (s1(1)>0)
    nib=nib+1;

```

```

        ibegin(nib)=1;
    end
    for i=1:length(s1)-1
        if (s1(i)==0) && (s1(i+1)>0)
            nib=nib+1;
            ibegin(nib)=i+1;
        end
    end
    iend=zeros(0);
    nie=0;
    for i=2:length(s1)
        if (s1(i-1)>0) && (s1(i)==0)
            nie=nie+1;
            iend(nie)=i-1;
        end
    end
    if (s1(length(s1))>0)
        nie=nie+1;
        iend(nie)=length(s1);
    end
    coord=zeros(0);
    ii=0;
    for k=1:nib
        fr_input=input(ibegin(k):iend(k),:);
        for i=ibegin(k):iend(k)
            s2=sum(fr_input,1);
            njb=0;
            jbegin=zeros(0);
            if s2(1)>0
                njb=njb+1;
                jbegin(njb)=1;
            end
            for j=1:length(s2)-1
                if (s2(j)==0) && (s2(j+1)>0)
                    njb=njb+1;
                    jbegin(njb)=j+1;
                end
            end
            nje=0;
            jend=zeros(0);
            for j=2:length(s2)
                if (s2(j-1)>0) && (s2(j)==0)
                    nje=nje+1;
                    jend(nje)=j-1;
                end
            end
            if s2(length(s2))>0
                nje=nje+1;
                jend(nje)=length(s2);
            end
        end
        for i=1:njb
            ii=ii+1;
            coord(ii,1)=ibegin(k);
            coord(ii,2)=iend(k);
            coord(ii,3)=jbegin(i);
            coord(ii,4)=jend(i);
        end
    end
    for i=1:ii
        i1=coord(i,1);
        i2=coord(i,2);
        j1=coord(i,3);
    end

```



```

        j2=coord(i,4);
        subIm_input{i}.img=input(i1:i2,j1:j2);
        subIm_input{i}.coord=[i1 i2 j1 j2];
    end

end

```

% Программы создания модель представления и формирования ТО

```

clc;
clear all;
close all;
% FOR SIZE 14

%% Define width and high of all symbol in anphabet.mat
load anphabet_s14.mat;
mat_width=zeros(0);
mat_high=zeros(0);
for k=1:length(anphabet)
    sym=anphabet{k};
    [nr nc]=size(sym);
    mat_width(k)=nc;
    mat_high(k)=nr;
end
% choose high of text region
h=max(mat_high);
% resize all symbol
for k=1:length(anphabet)
    [nr nc]=size(anphabet{k});
    temp=zeros(h,nc);
    dr=floor((h-nr)/2);
    temp(dr+1:dr+nr,:)=anphabet{k};
    new_anpha{k}=temp;
end
%% Define characteristics of each symbols
num_char=length(anphabet)/36;
anpha=['a' 'b' 'c' 'd' 'e' 'f' 'g' 'h' 'i' 'j' 'k' 'l' 'm' 'n' 'o' 'p' 'q'
'r' 's' 't' 'u' 'v' 'w' 'x' 'y' 'z' '0' '1' '2' '3' '4' '5' '6' '7' '8' '9'];
for k=1:36
    wk=0;
    hk=0;
    beta_wh=0;
    psym=0;
    max_w=0;
    max_h=0;
    for num=1:num_char
        nk=(num-1)*36+k;
        %
        wk=wk+mat_width(nk);
        hk=hk+mat_high(nk);
        %
        beta_wh=beta_wh+mat_width(nk)/mat_high(nk);
        %
        sym=anphabet{nk};
        [sr sc]=size(sym);
        psym=psym+sum(sum(sym))/(sr*sc);
        %
        if max_w<mat_width(nk)
            max_w=mat_width(nk);
        end
        if max_h<mat_high(nk)
            max_h=mat_high(nk);
        end
    end
end

```

```

        end
    end
    temp=zeros(h,max_w);
    for num=1:num_char
        nk=(num-1)*36+k;
        sym=new_anpha{nk};
        [sr sc]=size(sym);
        for r=1:sr
            for c=1:sc
                temp(r,c)=temp(r,c)+sym(r,c);
            end
        end
    end
end

temp_real=zeros(max_h,max_w);
for num=1:num_char
    nk=(num-1)*36+k;
    sym_real=anphabet{nk};
    [sr sc]=size(sym_real);
    for r=1:sr
        for c=1:sc
            temp_real(r,c)=temp_real(r,c)+sym_real(r,c);
        end
    end
end

charac{k}.anpha=anpha(k);
charac{k}.w=max_w;
charac{k}.h=max_h;
charac{k}.beta_wh=beta_wh/num_char;
charac{k}.psym=psym/num_char;
charac{k}.pmat_sym=temp/num_char;
charac{k}.pmat_real_sym=temp_real/num_char;
end
%% Define characteristics of each group from anphabet
% beta_wh_gr,charac{k}.group
gr1=[1;3;5;13;14;15;18;19;21;22;23;24;26];
gr2=[2;4;6;8;9;11;12;20;27;28;29;30;31;32;33;34;35;36];
gr3=[7;16;17;25];
gr4=[10];
group{1}.char=gr1;
group{2}.char=gr2;
group{3}.char=gr3;
group{4}.char=gr4;
gr1_wh=zeros(0);
gr2_wh=zeros(0);
gr3_wh=zeros(0);
gr4_wh=zeros(0);

gr1_h=zeros(0);
gr2_h=zeros(0);
gr3_h=zeros(0);
gr4_h=zeros(0);

for k=1:36
    for g1=1:length(gr1)
        if gr1(g1)==k
            charac{k}.group=1;
        end
    end
    for g2=1:length(gr2)
        if gr2(g2)==k

```

```

        charac{k}.group=2;
    end
end
for g3=1:length(gr3)
    if gr3(g3)==k
        charac{k}.group=3;
    end
end
for g4=1:length(gr4)
    if gr4(g4)==k
        charac{k}.group=4;
    end
end
end
end

g1=0;
g2=0;
g3=0;
g4=0;
for k=1:36
    if charac{k}.group==1
        g1=g1+1;
        gr1_h(g1)=charac{k}.h;
        gr1_w(g1)=charac{k}.w;
        gr1_wh(g1)=charac{k}.beta_wh;
    end
    if charac{k}.group==2
        g2=g2+1;
        gr2_h(g2)=charac{k}.h;
        gr2_w(g2)=charac{k}.w;
        gr2_wh(g2)=charac{k}.beta_wh;
    end
    if charac{k}.group==3
        g3=g3+1;
        gr3_h(g3)=charac{k}.h;
        gr3_w(g3)=charac{k}.w;
        gr3_wh(g3)=charac{k}.beta_wh;
    end
    if charac{k}.group==4
        g4=g4+1;
        gr4_h(g4)=charac{k}.h;
        gr4_w(g4)=charac{k}.w;
        gr4_wh(g4)=charac{k}.beta_wh;
    end
end
end
beta_wh_gr(1,:)=[min(gr1_wh) max(gr1_wh)];
beta_wh_gr(2,:)=[min(gr2_wh) max(gr2_wh)];
beta_wh_gr(3,:)=[min(gr3_wh) max(gr3_wh)];
beta_wh_gr(4,:)=[min(gr4_wh) max(gr4_wh)];
H_gr(1)=sum(gr1_h)/length(gr1);
H_gr(2)=sum(gr2_h)/length(gr2);
H_gr(3)=sum(gr3_h)/length(gr3);
H_gr(4)=sum(gr4_h)/length(gr4);
W_gr(1)=sum(gr1_w)/length(gr1);
W_gr(2)=sum(gr2_w)/length(gr2);
W_gr(3)=sum(gr3_w)/length(gr3);
W_gr(4)=sum(gr4_w)/length(gr4);

%% Define h,w, pmat, psym for each group
for k=1:4
    if k==1
        gr_w=gr1_w;
        group{k}.h=H_gr(k);
    end
end

```

```

end
if k==2
    gr_w=gr2_w;
    group{k}.h=H_gr(k);
end
if k==3
    gr_w=gr3_w;
    group{k}.h=H_gr(k);
end
if k==4
    gr_w=gr4_w;
    group{k}.h=H_gr(k);
end
wmin=min(gr_w);
wmax=max(gr_w);

temp1=zeros(0);
for nt=wmin:wmax
    ntl=nt-wmin+1;
    temp1(ntl,1)=nt;
    ntemp=0;
    for ig=1:length(gr_w)
        if gr_w(ig)==nt
            ntemp=ntemp+1;
        end
    end
    temp1(ntl,2)=ntemp;
end
wtemp=zeros(0);
iw=0;
for nt=1:size(temp1,1)
    if temp1(nt,2)>0
        iw=iw+1;
        wtemp(iw)=temp1(nt,1);
    end
end
group{k}.w=wtemp;
end
%
for g=1:4
    nc=length(group{g}.char);
    mat_psym=zeros(0);
    for ic=1:nc
        k=group{g}.char(ic);
        mat_psym(ic)=charac{k}.psym;
    end
    group{g}.mat_psym=mat_psym;
end
%
for g=1:4
    nw=length(group{g}.w);
    w_gr=group{g}.w;
    h=round(group{g}.h);
    for iw=1:nw
        w=w_gr(iw);
        temp=zeros(h,w);
        itemp=0;
        for ig=1:length(group{g}.char)
            ichar=group{g}.char(ig);
            if charac{ichar}.w==w
                itemp=itemp+1;
                rc=charac{ichar}.h;
                cc=charac{ichar}.w;
            end
        end
    end
end

```

```

        for irc=1:rc
            for jcc=1:cc
temp(irc,jcc)=temp(irc,jcc)+charac{ichar}.pmat_real_sym(irc,jcc);
                end
            end
        end
        end
        gtemp{iw}.pmat=temp/itemp;
        gtemp{iw}.w=w;
        gtemp{iw}.h=h;
    end
    if g==1
        group1=gtemp;
    end
    if g==2
        group2=gtemp;
    end
    if g==3
        group3=gtemp;
    end
    if g==4
        group4=gtemp;
    end
end

%% Define characteristics of all symbols
for k=1:36
    prob_sym(k)=charac{k}.psym;
end
psymbol=[min(prob_sym) max(prob_sym)];

%% Geration text region (TR)
% Variants for high TR - High_TR
h1=group{1}.h;
h2=group{2}.h;
h3=group{3}.h;
h4=group{4}.h;
High_TR=[h1 h2 h3 h4];

% Variant for width TR - Width_TR
% for h1 --> variants Wh1_TR
Wh1_TR=group{1}.w;
Width_TR{1}=Wh1_TR;
% for h2 --> variants Wh2_TR
Wh2_TR=group{2}.w;
Width_TR{2}=Wh2_TR;
% for h3 --> variants Wh3_TR
Wh3_TR=group{3}.w;
Width_TR{3}=Wh3_TR;
% for h4 --> variants Wh4_TR
Wh4_TR=group{4}.w;
Width_TR{4}=Wh4_TR;

% Matrix prob for TRrand
np=0;
for ih=1:4
    hTR=High_TR(ih);
    lenw=length(Width_TR{ih});
    if ih==1
        tempgr=group1;
    end
end

```

```

    if ih==2
        tempgr=group2;
    end
    if ih==3
        tempgr=group3;
    end
    if ih==4
        tempgr=group4;
    end
    for iw=1:lenw
        np=np+1;
        pmat_TR{np}=tempgr{iw}.pmat;
    end
end

%% *****
% GENERATE TO
%% *****
for raz=1:10
    nTR=0;
    % quantity of TR
    for ih=1:4
        hTR=High_TR(ih);
        lenw=length(Width_TR{ih});
        a=min(group{ih}.mat_psym);
        b=max(group{ih}.mat_psym);
        for iw=1:lenw
            nTR=nTR+1;
            wTR=Width_TR{ih}(iw);
            TRrand{nTR}=a+(b-a).*rand(hTR,wTR);
            % size(TRrand{nTR})
        end
    end
    % Generation TR with condition
    nk=nTR/np;
    for n=1:nk
        for k=1:np
            kk=(n-1)*np+k;
            [row col]=size(TRrand{kk});
            res=zeros(row,col);
            tempTR=TRrand{kk};
            temp_pTR=pmat_TR{k};
            for ir=1:row
                for ic=1:col
                    if tempTR(ir,ic)>=temp_pTR(ir,ic)
                        res(ir,ic)=1;
                    end
                end
            end
            TR{kk}=res;
        end
    end
end
% Show result - variant2
w_res=0;
mat_j=round(1+11.*rand(1,12));
len_res=12; % LEN TO
for i=1:len_res
    ii=mat_j(i);
    w_res=w_res+size(TRrand{ii},2)+1;
end
h_res=18;
res_generate=zeros(h_res,w_res);

```

```

jk=0;
for j=1:len_res
    jj=mat_j(j);
    if j==1
        jk1=1;
        jk2=jk1+size(TR{jj},2)-1;
        jk=jk2+1;
    else
        jk1=jk+1;
        jk2=jk1+size(TR{jj},2)-1;
        jk=jk2+1;
    end
    delta_ik=h_res-size(TR{jj},1);
    ik1=round(delta_ik/2);
    ik2=ik1+size(TR{jj},1)-1;
    res_generate(ik1:ik2,jk1:jk2)=TR{jj};
end
figure,imshow(res_generate);
RESULT{raz}=res_generate;
end
hr=0;
wr=0;
for r=1:raz
    [h w]=size(RESULT{r});
    hr=hr+h;
    if wr<=w
        wr=w;
    end
end
hr=hr+2*raz;
wr=wr+1;
image=zeros(hr,wr);
hh=2;
for r=1:raz
    [h w]=size(RESULT{r});
    image(hh:h+hh-1,2:w+1)=RESULT{r};
    hh=h+hh+2;
end
figure,imshow(image);

```

**% Программы создания набора обучения сверточных нейронных сетей для
% локализации ТО на полутоновых изображениях**

close all;

```
clear all;
clc;
```

```

num=0;
load data_GrayImg_MTO_X2.mat;
for i=1:length(data_GrayImg_MTO)
    num=num+1;
    imdata{num}=im2double(data_GrayImg_MTO{i});
    label(num)=1;
end
load data_GrayImg_piecesTO_X2.mat;
for i=1:length(data_GrayImg_piecesTO)
    num=num+1;
    imdata{num}=im2double(data_GrayImg_piecesTO{i});
    label(num)=1;
end
load dataRGB_nontext_7August.mat;

```

```

for i=1:length(dataRGB_nontext)
    num=num+1;
    BWnontext=rgb2gray(dataRGB_nontext{i});
    imdata{num}=2*(im2double(BWnontext))-1;
    label(num)=-1;
end

```

**% Программы создания набора обучения сверточных нейронных сетей для
% локализации ТО на цветных изображениях**

```

close all;
clear all;
clc;

num=0;
load dataRGB_IMTO_28Sep.mat;
for i=1:length(dataRGB_IMTO)
    num=num+1;
    imdata{num}=im2double(dataRGB_IMTO{i});
    label(num)=1;
end
load dataRGB_piecesTO_28Sep.mat;
for i=1:length(dataRGB_piecesTO)
    num=num+1;
    imdata{num}=im2double(dataRGB_piecesTO{i});
    label(num)=1;
end
load dataRGB_nontext_7August.mat;
for i=1:length(dataRGB_nontext)
    num=num+1;
    imdata{num}=im2double(dataRGB_nontext{i});
    label(num)=-1;
end

```


Приложение Б

Документи, подтверждающие внедрение результатов работы

ДИСКРЕТ
НАУКОВО-ВИРОБНИЧЕ ОБ'ЄДНАННЯ

65058, м. Одеса, а/с №18 тел: +38(048)733-68-60, факс: +388(048) 73-73-412
Сайт: www.diskret.com.ua E-mail: info@diskret.com.ua ЄДРПОУ 34381774

Акт
впровадження результатів дисертаційної роботи

В останні роки відбувається зростання кількості досліджень, які стосуються переробки графічної інформації, зокрема розпізнавання тексту.

Результати дисертаційної роботи Нгуєн Тхі Кхань Тієн «Інформаційна технологія локалізації текстових областей на зображеннях зі складним фоном», а саме метод локалізації текстових областей на зображеннях зі складним фоном на основі згорткової нейронної мережі та інформаційна технологія локалізації текстових областей були використані для моделювання та розробки програмно-алгоритмічного забезпечення при автоматизації обробки відеоінформації на підприємстві, а саме для модифікації системи автоматизації перевірки якості упаковки продукції та правильності розподілення її при формуванні партій товару.

Проведення автоматизованої перевірки якості упаковки продукції та правильності її розподілення при формуванні партій товару дозволило підвищити ефективність функціонування складу продукції на підприємстві.

Акт видано у зв'язку з захистом кандидатської дисертації і він не є основою для фінансових розрахунків.

Акт складено

Від ТОВ «НВО» ДІСКРЕТ
Генеральний директор



Г.С.Стаценко

Від ОНПУ
Завідувач кафедри інформаційних систем,
професор

С.Г.Антощук

«ЗАТВЕРДЖУЮ»

Проректор з науково-педагогічної
та виховної роботи Одеського
національного політехнічного
університету


«25» _____ Нестеренко С.А.
2015 р.



АКТ

про впровадження в навчальний процес результатів дисертаційної роботи за темою «Інформаційна технологія локалізації текстових областей на зображеннях зі складним фоном» на здобуття наукового ступеня кандидата технічних наук Нгуєн Тхі Кхань Тієн.

Комісія у складі професора кафедри інформаційних систем д.т.н. Крилова В.М., професора кафедри інформаційних систем д.т.н. Арсірій О.О., заступника завідувача кафедри інформаційних систем доцента к.т.н. Бабілонга О.Ю. склали цей акт про те, що у Одеському національному політехнічному університеті, при викладанні дисциплін «Методи та засоби комп'ютерних інформаційних технологій», «Методи і засоби штучного інтелекту» та «Обчислювальний інтелект», а також при виконанні курсових і дипломних робіт для студентів напряму підготовки 6.050101 «Комп'ютерні науки» та спеціальності 7.05010101 «Інформаційні управляючі системи та технології», використовувались наступні оригінальні матеріали, які отримано у дисертаційній роботі Нгуєн Тхі Кхань Тієн: ймовірнісна модель символу тексту, метод формування текстових областей з використанням ймовірнісної моделі символу тексту, кореляційно-екстремальний метод виявлення текстових областей з використанням згорткової нейронної мережі.

Професор кафедри ІС
професор, д. т. н.


В.М. Крилов

Професор кафедри ІС
доцент, д. т. н.


О. О. Арсірій

Заступник завідувача кафедри ІС
доцент, к. т. н.


О. Ю. Бабілонга