Федеральное государственное бюджетное учреждение науки
Институт системного программирования им. В.П. Иванникова
Российской академии наук

на правах рукописи

Обыденков Дмитрий Олегович

Методы противодействия анонимности при утечках текстовых документов посредством цифровых водяных знаков

Специальность 2.3.5 – математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
к.т.н., Маркин Юрий Витальевич

Москва – 2024

Оглавление

Введение

Использование информационных систем в коммерческих и государственных организациях приносит значительные выгоды, однако с их внедрением возникают новые угрозы, в частности, угрозы утечки информации. Данные о финансах, технологиях, сотрудниках и клиентах крайне важны для организаций, утечка подобных сведений может нанести серьезный финансовый и репутационный ущерб. Отчеты по инцидентам информационной безопасности подтверждают рост числа утечек. Большинство инцидентов связано с «инсайдерами» — сотрудниками компаний, действующими в сговоре с внешними нарушителями. Согласно исследованию InfoWatch в России доля утечек из-за внутренних нарушителей достигает 79% от общего числа инцидентов.

Для защиты от утечек используются Data Leakage Prevention (DLP) системы, которые могут работать в режиме реального времени или предотвращать возможные утечки превентивно. DLP-системы ориентированы на предотвращение утечек данных через сетевые каналы и не способны эффективно защищать аналоговые каналы утечек. К аналоговым сценариям утечек относятся фотографирование выведенного на экран документа или печать документа с последующей оцифровкой за пределами защищаемого контура при помощи сканера или фотоаппарата.

Методы противодействия утечкам информации через фотографирование документов на экране и/или их распечатку делятся на организационные и технические. Технические меры обычно подразумевают нанесение на документы цифровых водяных знаков (ЦВЗ) различных типов. ЦВЗ могут содержать информацию, позволяющую деанонимизировать пользователя, ставшего причиной утечки данных. Водяные знаки документов могут быть как явными, так и малозаметными или невидимыми для обнаружения и считывания скрытой информации

невооруженным глазом. Методы внедрения малозаметных ЦВЗ представляют особый интерес, поскольку они минимально влияют на удобство работы пользователей с документами.

Разработка методов внедрения ЦВЗ сопровождается поиском баланса между незаметностью изменений в документе, информационной емкостью и устойчивостью к искажениям. Классические подходы, работающие в домене преобразований (например, дискретное косинусное или быстрое преобразование Фурье), либо вносят слишком заметные изменения в изображения документов, либо недостаточно устойчивы к искажениям, возникающим в аналоговых каналах утечек. Вместе с этим, количество внедряемой в ЦВЗ информации как правило уменьшается при изменении параметров метода в сторону повышения незаметности или устойчивости. Разработка методов внедрения ЦВЗ, сочетающих в себе характеристики, позволяющие эффективно решать практические задачи, является актуальной задачей для исследователей в области стеганографии.

Применительно к текстовым документам перспективными являются структурные методы внедрения ЦВЗ. Существующие решения, в том числе EveryTag, используют структурные ЦВЗ для деанонимизации утечек посредством напечатанных документов, однако, не предполагают возможность работы в так называемом слепом сценарии – для извлечения внедренной информации требуется наличие оригинала документа. Необходимость хранения оригинальных документов накладывает значительные ограничения (в том числе, создание и поддержку единой базы конфиденциальных документов) на применимость подобных методов, поэтому исследование и разработка методов внедрения ЦВЗ с возможностью слепого извлечения встроенной информации, нацеленных на предотвращение анонимных утечек текстовых документов через распечатанные копии, является актуальной задачей.

Методы нанесения ЦВЗ для защиты документов, отображаемых на экране, делятся на динамические и статические. Динамические методы характеризуются перестроением водяного знака для адаптации под содержимое экрана, что может потребовать значительных вычислительных ресурсов системы, а также вызывать повышенную утомляемость пользователей из-за частых изменений на экране. Статические методы знаки демонстрируют меньшую заметность по метрикам PSNR/SSIM, но обладают низкой устойчивостью к искажениям, возникающим при фотографировании экрана. Для практического применения ЦВЗ также должен быть устойчивым к передаче изображения через мессенджеры, то есть сохранять информацию о распространителе при перекодировании и уменьшении размера изображения. Протестировать существующие на рынке коммерческие системы невозможно, а заявляемые в них характеристики ЦВЗ не подкреплены научными публикациями. Разработка методов нанесения водяных знаков, обеспечивающих низкую заметность для комфортной работы и высокую устойчивость к искажениям при утечках фотографий экрана с выведенным конфиденциальным документом, является актуальной задачей.

Степень разработанности темы. Отечественные и зарубежные исследователи публиковали работы в области внедрения ЦВЗ в контейнеры различных доменов, в том числе текст, изображения, аудио, видео. Методы, работающие в домене преобразований, развивали такие ученые, как I. Cox, T. Furon, A. Pramila, P. Dong и др. Внедрение водяных знаков в пространственную область представлено в работах J. Brassil, S. Low, N. Maxemchuk, M. Topkara, Y. Kim, А.А. Грушо, В.О. Писковским, Д.А. Семинихиным и др. За последние несколько лет появились нейросетевые методы, в частности, в работах J. Zhu, M. Tancik, W. Zhang, P. Fernandez и др.

Целью диссертационной работы является разработка методов противодействия анонимности при утечках текстовых документов посредством ЦВЗ со слепым декодированием, обеспечивающих устойчивость к искажениям, возникающим при печати или фотографировании отображаемых на экране документов с последующей передачей изображения через мессенджеры, а также имеющих визуальную незаметность и не вызывающих дискомфорта у пользователей.

Основные задачи:

1. Разработка архитектуры системы противодействия анонимности при утечках текстовых документов. Система должна обеспечивать внедрение в текстовые документы информации, позволяющей устанавливать виновников публичных утечек;
2. Разработка метода внедрения ЦВЗ в текстовые документы при печати, предполагающего слепое извлечение встроенной информации. Разработанный метод должен обладать устойчивостью к различным искажениям и преобразованиям, сопутствующим печати документа с последующей оцифровкой посредством сканирования или фотографирования. Внедренный в документ ЦВЗ должен быть визуально незаметен. Внедрение ЦВЗ не должно оказывать существенного влияния на скорость печати документов;
3. Разработка метода внедрения ЦВЗ в текстовые документы при выводе на экран, предполагающего слепое извлечение встроенной информации. Разработанный метод должен обладать устойчивостью к различным искажениям и преобразованиям, сопутствующим фотографированию выведенного на экран документа с последующей отправкой фотографии документа через мессенджер. Наличие ЦВЗ не должно вызывать дискомфорт у пользователей при использовании;

4. Реализовать систему противодействия анонимности при утечках текстовых документов с использованием разработанных методов и провести оценку их эффективности.

Научной новизной обладают следующие результаты работы:

1. Структурный метод внедрения ЦВЗ на основе сегментации текстового документа с помощью нейросетевых алгоритмов с возможностью слепого извлечения встроенной информаци, устойчивый к искажениям, возникающим при распечатывании и последующей оцифровке через фотографирование или сканирование, оптимизированный для работы на процессоре общего назначения с минимальным использованием вычислительных ресурсов;
2. Метод внедрения статических ЦВЗ, сгенерированных нейросетевым алгоритмом, в текстовые документы с возможностью слепого извлечения внедренной информации из фотографии экрана, устойчивый к алгоритмам сжатия изображений, применяемым в мессенджерах.

Теоретическая значимость диссертации заключается в разработке и усовершенствовании методов защиты текстовых документов от утечек информации через анонимные каналы с помощью ЦВЗ. В работе предложены новые решения, направленные на предотвращение несанкционированной передачи информации через печатные документы и отображаемые на экране, что расширяет научные представления в области внедрения ЦВЗ. Особую ценность представляют методы, которые обеспечивают эффективную защиту при условии, что документ может быть оцифрован после печати или сфотографирован с экрана, и при этом встроенная в документ информация сохраняется. Важной особенностью работы является использование нейросетевых алгоритмов для сегментации и внедрения информации в текстовые документы, что позволяет добиться

высокой устойчивости к искажениям, возникающим в процессе передачи изображения, и минимизировать визуальные изменения, что делает методы практически незаметными для пользователя. Предложенные решения являются актуальными в условиях современных угроз информационной безопасности, где важен баланс между эффективностью защиты и удобством использования информационных систем.

Практическая значимость. Разработаны и реализованы методы внедрения ЦВЗ в текстовые документы для защиты от утечек при фотографировании распечатанных или экранных копий. ЦВЗ малозаметны и не создают дискомфорта для пользователей, при этом позволяют деанонимизировать утечки через идентификатор сотрудника и устройства. Тестирование показало, что метод горизонтального смещения слов обеспечивает до 61.7% успешных извлечений для сканированных документов и 36.7% для фотографий, а метод перечеркивания — свыше 80% во всех сценариях. При наложении ЦВЗ на экран точность извлечения достигает 86.67% при непрозрачности цифрового водяного знака 8/255.

Реализованная система противодействия анонимности при утечках текстовых документов внедрена организацией ООО "СиТ" (акт о внедрении №612/0924 от 29.09.24).

Методология и методы исследования. В разработке и при тестировании алгоритмов внедрения ЦВЗ в текстовые документы был использован системный подход, основанный на моделировании угроз и нарушителя. Основные методы исследования включают анализ существующих решений, разработку и экспериментальное тестирование алгоритмов, а также математическое моделирование и статистическую обработку данных. В совокупности эти методы позволили объективно оценить эффективность созданных решений, а также их устойчивость к возможным угрозам.

Основные положения, выносимые на защиту:

1. Структурный метод внедрения ЦВЗ, предполагающий слепое извлечение внедренной информации, на основе сегментации изображения документа с помощью нейросетевого алгоритма, обладающего визуальной незаметностью и устойчивостью к искажениям, возникающим при распечатывании и последующей оцифровке посредством фотографирования или сканирования, и ориентированный под работу на процессоре общего назначения с минимальным потреблением вычислительных ресурсов;
2. Метод генерации ЦВЗ нейросетевым алгоритмом, предполагающий слепое извлечение внедренной информации и обладающий свойствами визуальной незаметности и устойчивости к искажениям, возникающим при фотографировании экрана и сжатии алгоритмами, применяемым в мессенджерах;
3. На основе предложенных методов реализована система противодействия анонимным утечкам текстовых документов, обеспечивающая внедрение уникальных идентификаторов сотрудников и используемых ими устройств в текстовые документы при печати и выводе на экран.

Апробация работы. Результаты работы обсуждались на следующих конференциях:

- Ежегодная научная конференция «Ломоносовские чтения», Москва, 20 – 29 апреля 2021 г.
- Международная конференция «Иванниковские чтения», Нижний Новгород, 24 – 25 сентября 2021 г.
- Научно-практическая Открытая конференция ИСП РАН им. В.П. Иванникова, Москва, 2 – 3 декабря 2021 г.
- Международная конференция «Иванниковские чтения», Казань, 23 – 24 сентября 2022 г.

- Научно-практическая Открытая конференция ИСП РАН им. В.П. Иванникова, Москва, 1 – 2 декабря 2021 г.
- Всероссийская конференция «Методы и технические средства обеспечения безопасности информации», Санкт-Петербург, 24 – 27 июня 2024 г.

Публикации и личный вклад автора. По теме диссертации опубликовано 8 научных работ, из них работы [8, 7, 17, 5, 3] опубликованы в журнале, входящем в список ВАК. Работа [65] опубликована в научном журнале, индексируемом системами Web of Science и Scopus. Получено 5 свидетельств регистрации программ для ЭВМ (Приложение А).

В работах [8, 9] автором представлена архитектура разработанной системы. В работах [7, 2] автором лично предложены методы внедрения структурных ЦВЗ, применимых к текстовым документам при печати. В статье [65] автору принадлежит методика коррекции ошибок при извлечении информации из ЦВЗ на фотографии текстового документа, выведенного на экран. В работе [17] автором выполнен обзор существующих методов внедрения ЦВЗ в текстовые документы. В статье [5] описана разработанная автором методика тестирования методов нанесения водяных знаков при печати, приближенная к условиям эксплуатации. В работе [3] автором собран набор данных для тестирования и предложен набор преобразований для имитации искажений, возникающих при оцифровке посредством сканирования и фотографирования распечатанных копий текстовых документов.

Структура и объем диссертационной работы. Диссертация состоит из введения, пяти глав и заключения, изложенных на 164 страницах, списка литературы из 67 наименования, содержит 48 рисунков и 24 таблицы.

Глава 1. Обзор предметной области

Первая глава состоит из пяти разделов. В разделе 1.1 приведено описание классических подходов борьбы с утечками, используемых на сегодняшний день в специалистами по информационной безопасности.

Раздел 1.2 полностью посвящен обзору представленных на рынке решений по обеспечению информационной безопасности, нацеленных на противодействие утечками конфиденциальных документов посредством фотографирования или сканирования распечатанных копий, а также посредством фотографирования экрана. Информация о решениях получена из открытых источников.

В разделе 1.3 приведен обзор опубликованных методов и техник внедрения ЦВЗ в документы, особый интерес представляли методы, устойчивые к искажениям при печати. Рассмотрены методы внедрения ЦВЗ, работающих в домене преобразований для встраивания информации, структурные и лингвистические методы. Раздел 1.4 включает описание представленных в литературе методов внедрения ЦВЗ, устойчивых к искажениям при фотографировании выведенных на экран документов.

Раздел 1.5 содержит выводы по первой главе.

1.1 Защита от утечек информации при помощи DLP-систем

Для борьбы с утечками информации активно применяют системы класса Data Leakage Prevention (DLP), разделяемые по скорости реагирования на активные и пассивные. Решения первого типа позволяют предотвращать утечки в режиме реального времени, поскольку осуществляют непрерывный мониторинг периметра. Однако, такие системы могут ошибаться и блокировать допустимую активность пользователей, поскольку на практике сложно избежать ложно-положительных срабатываний. Поэтому в ряде случаев обосновано

использование пассивных DLP-систем, предназначенных для упреждающего обнаружения потенциальных утечек информации.

Рисунок 1.1. Примеры данных в различных состояниях.

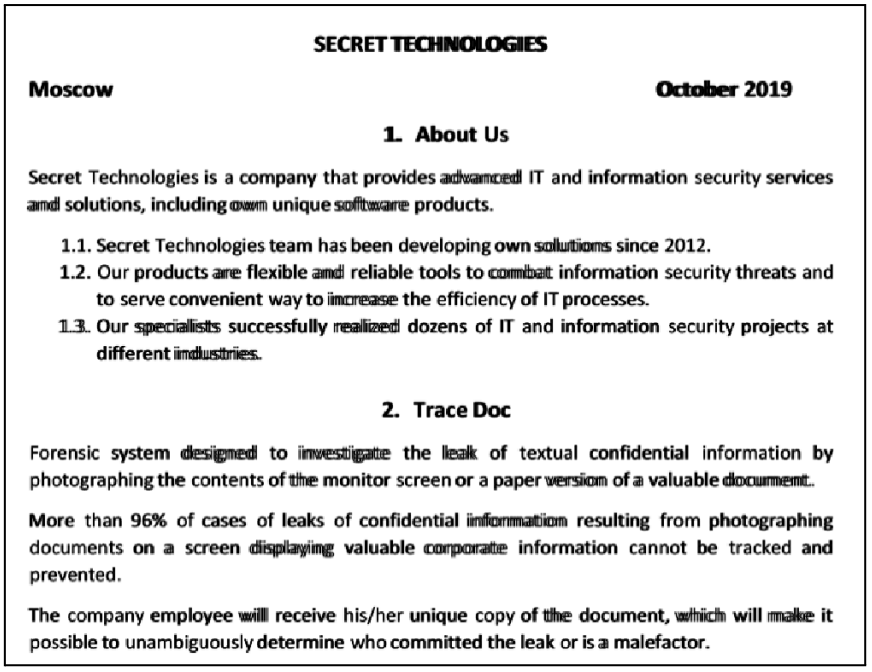
Компоненты DLP-системы могут размещаются в различных точках периметра и соответственно выполнять различные функции. Компоненты DLP, размещаемые на компьютере пользователя (data-in-use), именуются агентами и охватывают сразу несколько возможных каналов утечек. Агент контролирует потоки данных через локальные устройства ввода-вывода: обмен данными через сетевые интерфейсы, копирование файлов на внешние носители, отправка документов на печать и другие. Размещение агентов на рабочем месте позволяет обнаруживать подозрительную активность и предотвращать потенциальные утечки. Компоненты DLP-системы, размещаемые на сетевом шлюзе, контролируют содержимое сетевого трафика при помощи технологии DPI (Deep Packet Inspection). К компонентам данного типа предъявляются особые требования к производительности, поскольку пропускная способность сетей в корпоративных системах может достигать десятков гигабит в секунду. Требования к производительности сильно ограничивают сложность алгоритмов анализа, однако контроль на данном уровне чрезвычайно важен, поскольку позволяет обеспечить защиту от утечек с устройств, неподконтрольных администратору сети, например, с личных мобильных устройств или устройств, используемых в рамках концепции BYOD (Bring Your Own Device). Компоненты поиска размещаются (data-at-rest) рядом с хранилищами данных, содержащих

конфиденциальную информацию, и выполняют непрерывное сканирование ресурсов организации на предмет несанкционированной публикации документов, выполняя таким образом превентивную защиту от утечек. При эксплуатации таких систем особое значение имеют генерация и визуализация отчетов о событиях, необходимые для работы аналитика службы безопасности.

Обнаружение конфиденциального документа может выполняться как по формальным признакам (специальным атрибутам документа), так и на основе анализа содержимого. Зачастую для поиска по содержимому задаются нечеткие критерии соответствия, как например поиск по сигнатурам или регулярным выражениям. Более продвинутые методы опираются на лингвистический анализ содержимого или вычисляют цифровой отпечаток документа. И наконец, ряд систем использует OCR (Optical Character Recognition) [42][56] для поиска конфиденциальных документов.

1.2 Обзор существующих решений защиты документов при печати или выводе на экран

На рынке решений в Российской Федерации присутствует ряд решений, нацеленных на защиту каналов утечек конфиденциальных документов посредством фотографирования распечатанной копии или выведенных на экран. В данном разделе описаны решения от шести различных разработчиков решений данного типа. Информация о данных продуктах взята из публичных источников, таких как сайты разработчиков или партнеров. Опубликованные материалы носят маркетинговый характер и могут содержать неточности в описании характеристик продуктов.



SECRET TECHNOLOGIES

Moscow **October 2019**

1. About Us

Secret Technologies is a company that provides advanced IT and information security services and solutions, including own unique software products.

1.1. Secret Technologies team has been developing own solutions since 2012.
1.2. Our products are flexible and reliable tools to combat information security threats and to serve convenient way to increase the efficiency of IT processes.
1.3. Our specialists successfully realized dozens of IT and information security projects at different industries.

2. Trace Doc

Forensic system designed to investigate the leak of textual confidential information by photographing the contents of the monitor screen or a paper version of a valuable document.

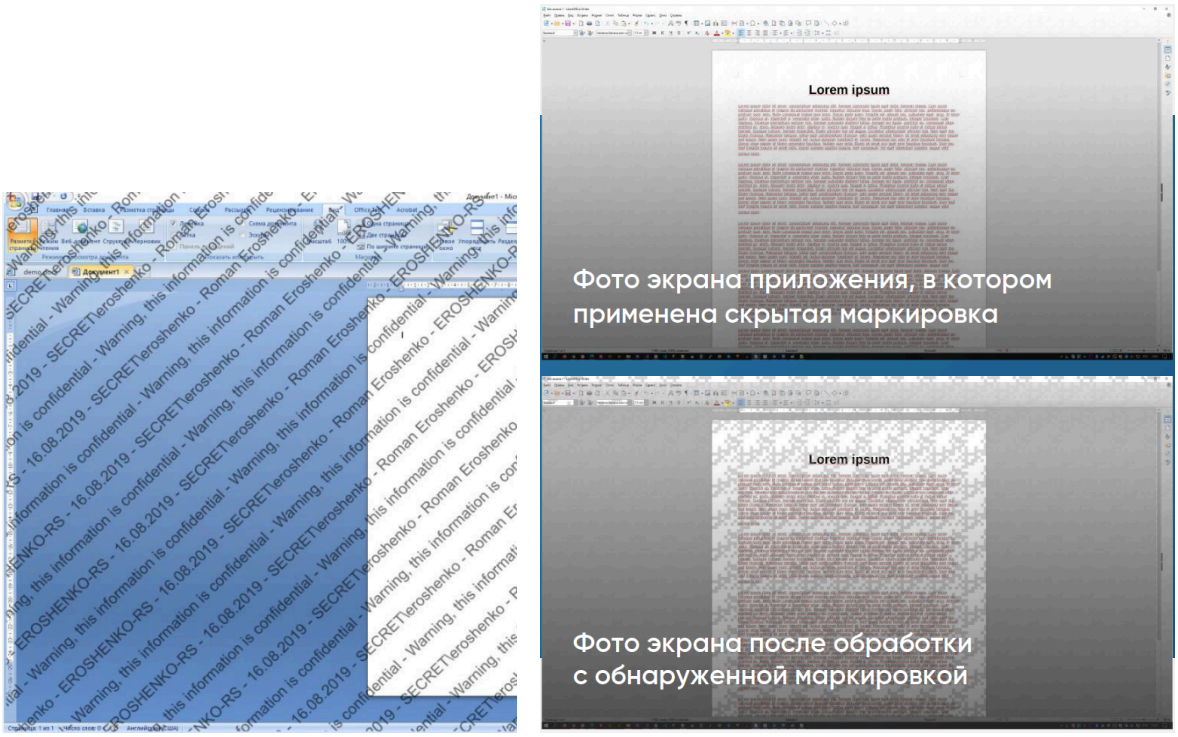
More than 96% of cases of leaks of confidential information resulting from photographing documents on a screen displaying valuable corporate information cannot be tracked and prevented.

The company employee will receive his/her unique copy of the document, which will make it possible to unambiguously determine who committed the leak or is a malefactor.

Рисунок 1.2. Демонстрация изменений документа при уникализации при помощи продукта TraceDoc. Изображение из презентации продукта.

Компания *SecretTechnologies* разрабатывает решение для защиты конфиденциальной информации от утечек и идентификации источника утечки информации. Решение *TraceDoc* [60] предназначено для проведения расследования факта утечки текстовой конфиденциальной информации путём фотографирования содержимого экрана монитора, или бумажной версии ценного документа. Программа позволяет определить с какого устройства была украдена информация по фото, видео, скриншотам документа после попадания изображения документа в открытые источники или передачи заинтересованным людям. При обнаружении утечки сотрудник ИБ компании проводит расследование инцидента — идентификацию уникализированной копии документа. Создание уникальной копии происходит при совершении сотрудником действий с документом в СЭД организации: просмотр, скачивание, печать или отправка по электронной почте. Уникализация документа осуществляется посредством сдвигов текстового содержимого. Преобразование не изменяет количество страниц и форматирование документа. Для

расследования требуется изображение документа, допускается попадание на фото части распечатанной страницы документа, фотографирование под углом и фотографирование с монитора.



(a) *(b)*

Рисунок 1.3. Демонстрация наносимых на экран цифровых водяных знаков: видимых (a) и скрытых (b) при помощи продукта ScreenGuard. Изображение из презентации продукта.

Помимо *TraceDoc* компания *SecretTechnologies* имеет решение *ScreenGuard* [52] — систему снижения рисков утечки информации путем фотографирования с помощью накладываемых на окна приложений цифровых водяных знаков. Согласно опубликованным материалам, на АРМ сотрудника выполняется полное или частичное перекрытие окна идентификационной информацией. Накладываемые водяные знаки могут быть видимыми и включать настраиваемый набор выводимых атрибутов (имя пользователя, дата, IP-адрес и другие) или произвольное изображение, а также скрытыми. Расследование инцидента утечки

предполагается осуществлять по фотографии или скриншоту экрана с конфиденциальной информацией.

Помимо *TraceDoc* и *ScreenGuard*, существует решение *PrinterGuard* [45] от *SecretTechnologies*, предназначенное для мониторинга, повышения безопасности и экономии ресурсов при печати. Решение предоставляет функционал мониторинга и детализации процессов печати по различным параметрам, разграничение доступа, логирование событий печати и нанесение меток на документы (текстовых и графических). Система реализована на основе виртуального принтера через который проходят отправляемые на печать документы, печать документа начинается только после авторизации сотрудником непосредственно на принтере.

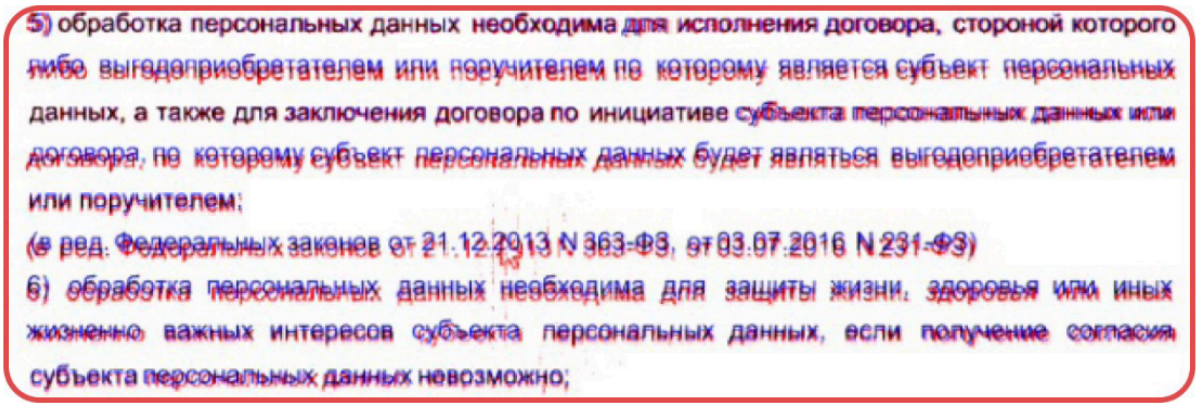


Рисунок 1.4. Демонстрация вносимых в документ изменений при помощи технологии EveryTag. Изображение из презентации продукта.

Компания *EveryTag* [28] также развивает технологию нанесения ЦВЗ для борьбы с утечками конфиденциальных документов, скомпрометированных в формате фото и скриншотов. Модуль встраивания ЦВЗ формирует уникальную копию для каждого пользователя при помощи механизма смещения текста в документах. Экспертиза позволяет определить источник утечки по фотографии, скриншоту или ксерокопии документа или фрагмента документа, допускается фотографирование под углом и порча (загрязнение и смятие) физической копии документа. Идентификация уникальной копии, ставшей субъектом утечки,

выполняется при помощи сопоставления с созданными копиями, однако на сервере хранятся только параметры преобразования данной копии. Уникализация документов осуществляется при выгрузке из СЭД, включении в почтовые вложения и отправке на печать. Внедрение ЦВЗ при печати реализовано через виртуальный принтер, который отправляет документ на сервер печати, выполняющий создание уникальной копии с заданными параметрами.

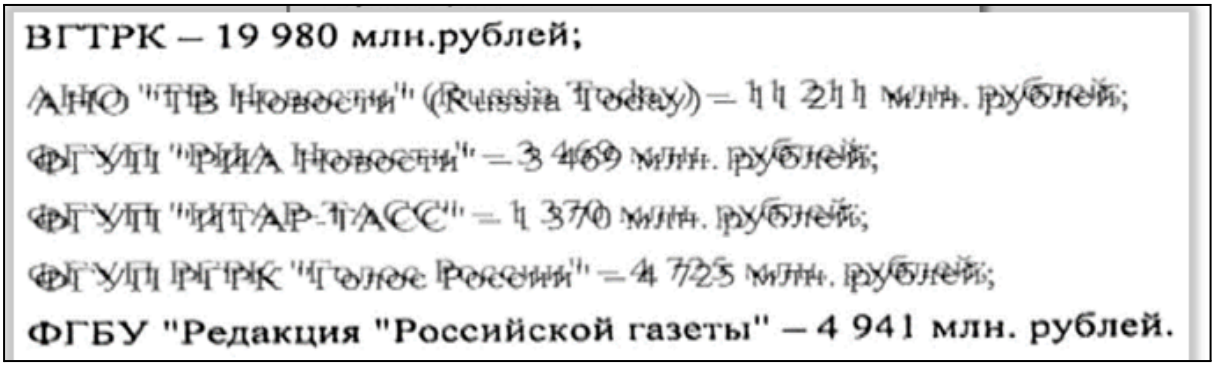
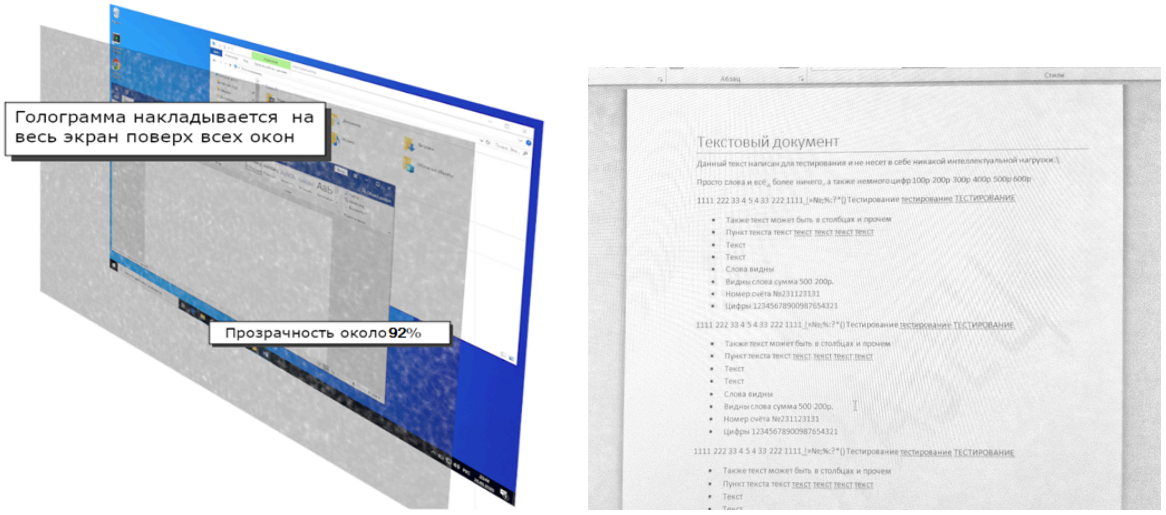


Рисунок 1.5. Демонстрация вносимых в документ изменений при помощи решения SafeCopy. Изображение из блога разработчика.

Платформа *SafeCopy* [50] от *НИИ СОКБ* нацелена на снижение рисков возможной утечки корпоративной информации, возникающих при распространении электронных и печатных копий документов. Решение позволяет определить источник утечки информации, если документ был сфотографирован, распечатан или отправлен по почте. Уникализация документа выполняется посредством механизма модификации текстовых элементов документов – смещение блоков текста, растяжение/сжатие, внесение незаметных артефактов и другое. Идентификация источника утечки осуществляется посредством наложения изображения утечки на изображение каждой сгенерированной копии.

Программный комплекс *«Виконт»* [12] предназначен для расследования инцидентов хищения информации, отображаемой на экране автоматизированных рабочих мест, с помощью фотофиксации. Система накладывает уникальный водяной знак на экран, кодируя идентификатор

АРМ и дату съемки. Этот метод защищает как статические, так и динамические изображения, и не имеет ограничений по формату выводимых на экран документов. Заявлено, что расследование возможно даже при наличии помех на снимках, съемке с большого расстояния (более 1 метра), под углом или по фрагменту экрана. Автономное обновление водяных знаков в течение 7 дней указывает на необходимость периодического взаимодействия с сервером для генерации меток. Скорость расследования зависит от числа инсталлированных агентов на АРМ в защищенной среде.



(a) *(b)*

Рисунок 1.6 – Демонстрация метода наложения водяного знака (a) и пример снимка выведенного на экран документа с наложенным водяным знаком (b) от ПК "Виконт". Изображение с сайта разработчика.

Платформа *Docs Security Suite (DSS)* [26] от *CrossTech Solutions Group* разработана для минимизации рисков компрометации конфиденциальных документов. Решение позволяет внедрять как видимые, так и скрытые цифровые водяные знаки, что дает возможность службе безопасности идентифицировать пользователя, допустившего утечку. Программа позволяет задавать документы атрибутами конфиденциальности, а агент, установленный на АРМ, контролирует

доступ к документам и предотвращает несанкционированное использование. Также реализована функция уникализации документов, обеспечивающая скрытое размещение идентификационной информации. Для файлов в формате Word информация кодируется изменением межбуквенных интервалов, представляя битовую последовательность длиной 38 бит. В PDF-документах уникализация выполняется с помощью алгоритма "Кружки", где информация кодируется через диаметр и яркость кружков, расположенных в межстрочных интервалах.

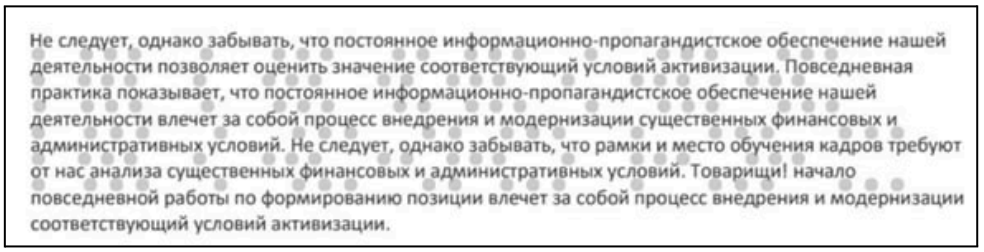

Не следует, однако забывать, что постоянное информационно-пропагандистское обеспечение нашей деятельности позволяет оценить значение соответствующий условий активизации. Повседневная практика показывает, что постоянное информационно-пропагандистское обеспечение нашей деятельности влечет за собой процесс внедрения и модернизации существенных финансовых и административных условий. Не следует, однако забывать, что рамки и место обучения кадров требуют от нас анализа существенных финансовых и административных условий. Товарищи! начало повседневной работы по формированию позиции влечет за собой процесс внедрения и модернизации соответствующий условий активизации.

Рисунок 1.7. Демонстрация метода наложения водяного знака при помощи алгоритма “Кружки” от CrossTech Solutions. Изображение из документации продукта.

Программный комплекс *reTributor* [48] от компании *MAS Platform* предназначен для автоматизации процесса определения источника утечки конфиденциальных документов за счет внедрения незаметных для пользователя цифровых водяных знаков. Данное решение позволяет организовать защищенное хранилище конфиденциальных документов. Персонифицированная уникальная копия документа создается и предоставляется авторизованному пользователю в электронном виде с возможностью печати на бумажном носителе, отправки по почте или через другие средства передачи документов. При встраивании ЦВЗ осуществляется визуальная деформация текстового и графического содержания документа без нарушения его структуры. Расследование инцидента утечки возможно при наличии распечатанного файла, электронной версии документа или его фотографии, скриншота или снимка экрана рабочего устройства на смартфон. Для проведения

расследования достаточно фрагмента – от 15% от всей страницы. Также решение имеет компонент, нацеленный на повышение безопасности при работе с мессенджерами. DLP-решение *reTributor Messenger* обеспечивает безопасный обмен документами между пользователями, заменяя исходные файлы ссылками на персонализированные версии документа.

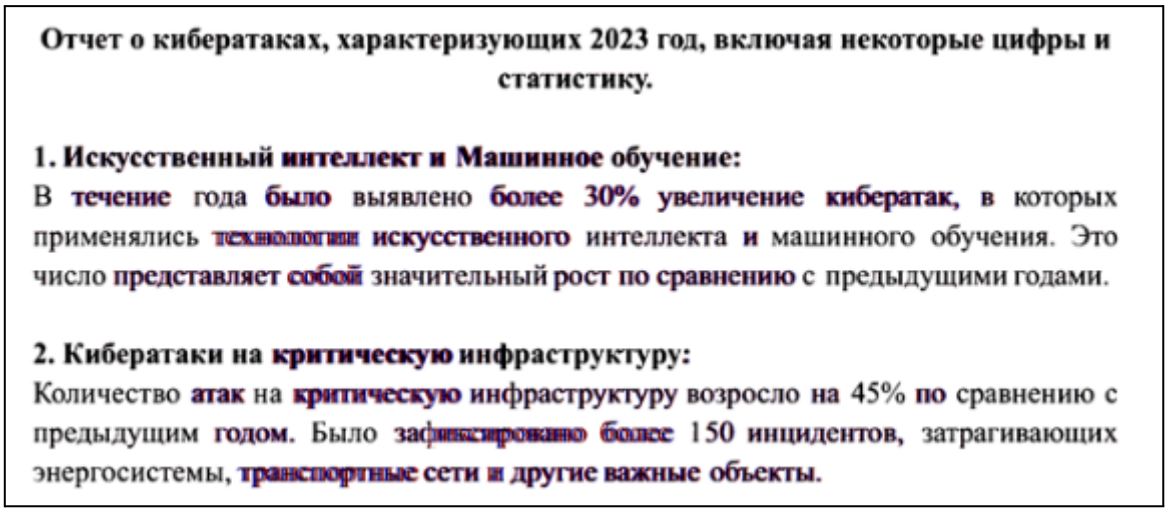


Рисунок 1.8. Демонстрация вносимых в документ изменений при помощи решения reTributor. Изображение из презентации продукта.

Описанные в данном разделе решения, представленные на рынке средств защиты информации, были проанализированы по следующим критериям и представлены в таблице 1.1:

- *Механизм* – краткое описание метода кодирования информации в документе посредством цифрового водяного знака;
- *Объекты утечки* – перечисление способов представления документа при которых возможно расследование инцидента утечки;
- *Создание цифрового водяного знака* – компонент системы, осуществляющий создание цифрового водяного знака;
- *Форматы документов* – перечисление форматов, в которых может быть представлен документ, для внедрения цифрового водяного знака.

Средства защиты информации, создающие скрытые ЦВЗ, можно разделить на две категории: одни изменяют структуру документа для

создания уникальной копии, другие накладывают водяной знак поверх отображаемого документа. Первая группа решений при расследовании утечек сопоставляет изображение утекшего документа со всеми ранее сгенерированными копиями. Этот подход требует централизованного защищенного хранилища оригиналов документов и информации о всех созданных копиях с водяными знаками. Если данных об уникальной копии документа в хранилище нет, то идентификация субъекта утечки становится невозможной. Решение *Docs Security Suite* использует изменение межбуквенных интервалов для кодирования информации, однако водяной знак на основе этого метода визуально более заметен, чем подходы, основанные на смещении слов.

Таблица 1.1. Сравнение существующих на рынке РФ решений.

Разработчик	**Продукт**	**Механизм**	**Объекты утечки**	**Создание цифрового водяного знака**	**Форматы документов**
Secret Technologies	*TraceDoc*	Горизонтальное смещение текстовых элементов	Фотография распечатанной копии, фотография или скриншот экрана	Модуль СЭД	?
	Screen Guard	Полупрозрачное окно с цифровым водяным знаком поверх окна приложения	Фотография или скриншот экрана	Локальный агент на АРМ	Без ограничений
EveryTag	*Модуль СЭД*	Вертикальное и горизонтальное смещение текстовых элементов	Фотография распечатанной копии, фотография или скриншот экрана	Модуль СЭД	PDF, DOCX, PPTX, ODT, RTF, PNG, JPG
	Модуль печати			ETP-сервер	XPS
	Модуль почты			Почтовый сервер	?
Виконт Секьюрити	*ПК Виконт*	Полупрозрачное окно с цифровым водяным знаком поверх поверх всех приложений	Фотография или скриншот экрана	Локальный агент на АРМ (автономно до 7 дней)	Без ограничений
НИИ СОКБ	*SafeCopy*	Смещение блоков текста, растяжение или сжатие, внесение артефактов	Фотография распечатанной копии, фотография/видео или скриншот экрана	Модуль СЭД	?
CrossTech	*Docs*	Изменение	Фотография	Локальный агент	PDF, DOCX

Solutions Group	*Security Suite (DSS)*	межбуквенных интервалов, кружки в межстрочных интервалах	распечатанной копии, фотография или скриншот экрана	на АРМ	(скрытые ЦВЗ)
MAS Platform	*reTributor*	Горизонтальное смещение текстовых элементов	Фотография распечатанной копии, фотография или скриншот экрана	Модуль СЭД	?

Рассмотренные решения, использующие полупрозрачное окно для наложения цифрового водяного знака поверх содержимого документа, не имеют ограничений на формат документов. Однако для методов этого типа особенно актуальны вопросы точности декодирования внедренной информации и заметности водяного знака. Поскольку эти решения являются коммерческими продуктами, проведение полноценного тестирования для объективной оценки их характеристик затруднено. Также, ПК "*Виконт*" имеет ограничение на время автономной работы без обновления водяного знака, что накладывает дополнительные ограничения на его применение.

1.3 Обзор методов внедрения ЦВЗ в документы при печати

Задача внедрения ЦВЗ в документы при печати широко освещена во множестве публикаций. Существует большое разнообразие методов, опирающихся как на *область преобразований* (transform domain), так и на *пространственную область* (spatial domain). Например, метод [27] Dong и др. (2005) может использоваться для защиты авторских прав на изображения при помощи водяных знаков, устойчивых к геометрическим искажениям. Перед внедрением ЦВЗ выполняется нормализация изображения для конвертации в состояние, инвариантного аффинным преобразованиям (растяжение, сжатие, поворот). Специальный алгоритм генерирует из заданной двоичной последовательности ЦВЗ, который впоследствии встраивается в среднечастотный диапазон *дискретного косинусного преобразования* (ДКП) нормализованного изображения. Далее,

к нормализованному изображению с встроенным ЦВЗ применяется операция обратная нормализации. Для извлечения внедренной информации к изображению применяются обратные операции: сначала изображение с ЦВЗ нормализуется, к нормализованному изображению применяется дискретное косинусное преобразование и уже из преобразованного изображения извлекается внедренная информация.

Авторы статьи оценили устойчивость данного метода внедрения ЦВЗ к разным видам искажений. Результаты тестирования показали, что метод устойчив к искажениям поворота, растяжения, изменению соотношения сторон, горизонтальному и вертикальному отражению и другим аффинным преобразованиям. Однако, данный метод показал низкую устойчивость к обработке изображения с ЦВЗ медианным фильтром и фильтром Гаусса, а также к JPEG сжатию — тестирование показало, что от 0.5% до 6.5% битов внедренной информации было декодированно неверно после атак.

В других работах частотный домен преобразования может использоваться для внедрения вспомогательной информации. Например, диссертации [44] Pramila 2018 года посвящена задаче внедрения ЦВЗ, устойчивых к искажениям, возникающим при распечатке изображений с последующей оцифровкой фотографированием. Дискретное преобразование Фурье (DFT) применяется для внедрения специального паттерна, необходимого для коррекции перспективы фотографии распечатанного изображения. Для встраивания информации в водяной знак используется вейвлет преобразование Хаара. В данной работе, как и в предыдущей, для встраивания информации используется диапазон средних частот. Это объясняется тем, что изменения в диапазоне низких частот приводят к существенным искажениям изображения, а изменения внесенные в область высоких частот хоть и являются незаметными для

глаза, но не сохраняются даже при незначительных модификациях изображения.

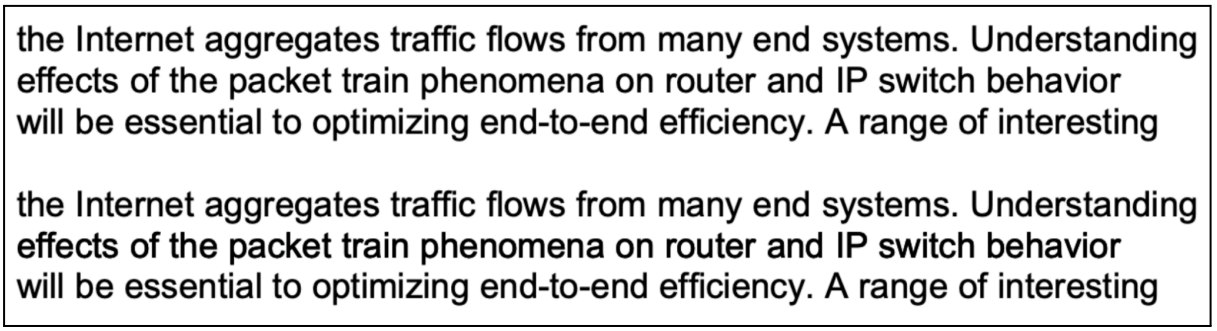


Рисунок 1.9. Пример внедрения информации путем смещения строк. Сверху оригинал, снизу центральная строка смещена вниз на 1/300 дюйма.

Методы внедрения водяных знаков в документы путем внесения структурных изменений в изображения документов публикуются с 1995 года, одними из первых над подобными методами работали исследователи Brassil, Low, Maxemchuk и др. Например, в статье [41] предлагается внедрять информацию путем смещения слов и строк в различных направлениях. Для этого авторы разработали эвристический способ разбиения документа на строки и слова путем анализа вертикального и горизонтального профиля изображения документа. После выделения строк и слов, в документе выбираются контрольные строки, которые в кодировании информации не используются. Остальные строки, которые находятся между контрольными, являются кодирующими и сдвигаются вверх или вниз, кодируя таким образом информацию. Слова в кодирующих строках объединяются в группы по три слова. Каждая такая группа кодирует информацию путем смещения среднего слова вправо или влево. Пример фрагмента изображения документа с внедренной информаций изображен на рисунке 1.9.

Также существуют структурные методы встраивания ЦВЗ в документы, использующие особенности написания символов алфавита. Например, в статье Shirali-Shahrez [54] при внедрении цифрового водяного знака используются точки в буквах, присутствующих в 15 из 28 букв

арабского алфавита. Информация кодируется путем сдвига одной или нескольких точек вверх (рисунок 1.10). Внедрение начинается с первой буквы, содержащей точки, и продолжается, пока вся бинарная последовательность не будет закодирована. В остальных буквах с точками, которые не участвовали в кодировании, точки смещаются случайным образом, чтобы не выделяться на фоне тех, в которых содержится закодированная информация. Первые несколько бит резервируются для кодирования величины смещения точек, что позволяет алгоритму извлечения различать случайные сдвиги от полезной нагрузки.

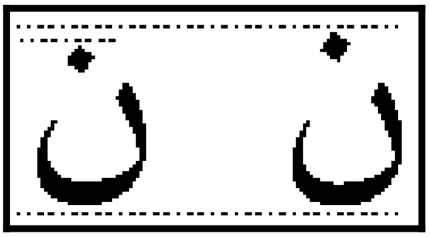
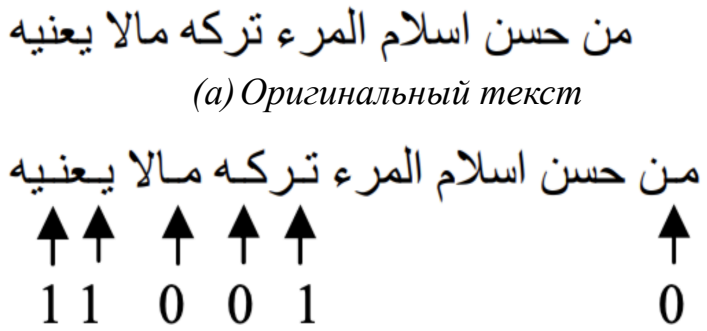


Рисунок 1.10. Пример смещения точки у буквы арабского языка.

В статье Gutub и др. [18] описан похожий метод внедрения ЦВЗ, различие в механизме кодирования информации на основе изменения длины штрихов определенных букв (рисунок 1.11). Данный метод, как и предыдущий, не изменяет смысла написанного.



(a) Оригинальный текст

(b) Измененный текст

Рисунок 1.11. Пример использования механизма кодирования информации посредством изменения длины штрихов тексте на арабском языке.

В статье Xiao и др. [63] описан метод внедрения ЦВЗ в текстовый документ путем изменения начертания глифов букв. Авторы составили

кодовую книгу для всех символов определенного набора шрифтов. В кодовой книге каждому глифу ставится в соответствие одна или несколько разных версий данного глифа, которые почти не отличимы от оригинала. При внедрении ЦВЗ текст делится на сегменты по пять символов. Каждый такой сегмент может кодировать разное количество бит, зависящее от суммарного количества вариантов символов конкретного сегмента, содержащихся в кодовой книге (пример на рисунке 1.12). Извлечение внедренной информации происходит путем определения варианта глифа в кодовой книге при помощи нейронной сети. Помимо этого, авторы используют коды коррекции ошибок, что гарантирует восстановление оригинального сообщения с частотой ошибок, не превышающей установленную. Главным недостатком данного метода является то, что он работает с ограниченным числом шрифтов, а поддержка новых шрифтов требует составления отдельных кодовых книг и ресурсоемкого обучения нейросетей.

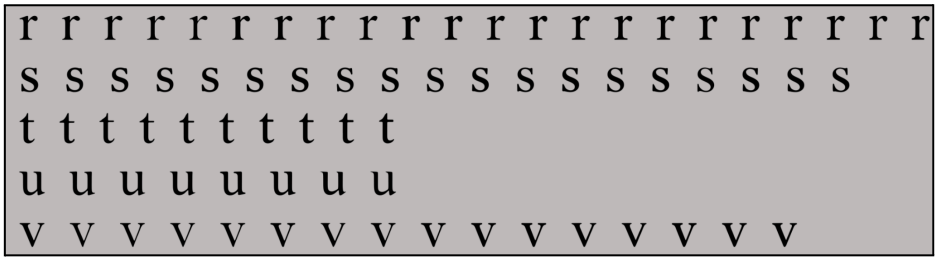


Рисунок 1.12. Пример части кодовой книги для некоторых символов.

Лингвистические методы внедрения цифровых водяных знаков в текст опираются на модификацию текста таким образом, чтобы эти изменения были незаметны для читателя. Семантические методы основаны на модификации семантических характеристик текста для внедрения информации. Например, в статье Topkara и др. [59] описан метод внедрения водяных знаков на уровне предложений, который использует парафразирование и изменение синтаксической структуры текста. Метод направлен на то, чтобы сделать водяной знак менее

заметным и одновременно устойчивым к изменениям текста. Водяные знаки внедряются через синтаксические перестановки и синонимическую замену, сохраняя смысл исходного текста. Синтаксические подходы предполагают модификацию текста без значительного изменения смысла или тональности содержимого текста. В различных языках могут использоваться синтаксические структуры или свойства, специфичные для определенных языков. В статье Shirali-Shahrez [53] описан метод, основанный свойствах буквы ﻻ (“лям”) в арабском языке, допускающий разные формы написания, что используется для кодирования информации. Также в настоящее время активно развиваются методы, основанные на использовании частотных характеристик, изменяющих частоты появления определенных слов или фраз, что может быть использовано для внедрения водяного знака. Основным преимуществом лингвистических методов является устойчивость атакам на водяные знаки: перепечатывание текста или использование OCR.

Рассмотрены различных методы и техники внедрения цифровых водяных знаков в документы, особый интерес представляли методы, устойчивые к искажениям при печати. Лингвистические методы обладают наибольшей устойчивостью, однако большинство методов обладают низкой емкостью и могут искажать содержимое текста, что накладывает значительные ограничения на практическое применение методов. Методы работающие в домене преобразований показывают хорошие метрики устойчивости и емкости водяного знака, но применимы в первую очередь к изображениям с переходами цвета (фотографии, иллюстрации). Использование данных методов на текстовых документам приводит к существенному снижению качества изображения текстового документа. Опубликовано множество структурных методов внедрения ЦВЗ, основанных на различных подходах кодирования информации. Структурные методы, основанные на высвечивании контуров символов

[57] или искажении шрифтов [63], предъявляют значительные требования к разрешению и качеству изображений для извлечения внедренной в ЦВЗ информации. Однако, методы, основанные на смещениях более крупных текстовых элементов как строки или слова, демонстрируют высокую устойчивость к искажениям и незаметность изменений, однако требуют использования алгоритмов поиска текстовых элементов на изображении документа.

1.4 Обзор методов внедрения ЦВЗ в документы при выводе на экран

Задача внедрения цифровых водяных знаков в документы, устойчивых к искажениям, возникающим при фотографировании с экрана, рассматривалась исследователями. Метод [33], предложенный Gugelmann и др. в 2018 году, основан на плавном изменении яркости областей на экране. Сообщение встраивается путем понижения или повышения яркости круговых областей, в зависимости от значения отдельных битов. Плавное изменение яркости незаметно для восприятия зрительной системой человека, но различимо цифровой камерой. На рисунке 1.13а представлен пример фотографии экрана с ЦВЗ. Метод обладает высокой емкостью встраивания — до 1000 бит на весь экран, но при такой емкости показывает высокую долю ошибки BER (Bit Error Rate) — до 25% битов извлекается неверно. Проблема решается при помощи избыточного кодирования встраиваемой информации, благодаря которому 40-битная последовательность извлекается без ошибок. Маска круговых областей, накладываемая на изображение на экране, не зависит от этого изображения и может быть сгенерирована на основе кодируемой бинарной последовательности заранее. Это позволяет использовать метод в режиме реального времени, поскольку маска остается постоянной. Метод был

протестирован авторами статьи на фотографиях экрана и на фотографиях после базовой обработки в графическом редакторе (уменьшение размеров изображения в 2 раза, изменение цветов, яркости, контрастности, баланса белого), но в ходе тестирования не было изучено влияние пространственного положения камеры и применения алгоритмов сжатия.

(a) Пример из статьи [33] *(b) Пример из статьи [17]* *(c) Пример из статьи [31]*

Рисунок 1.13. Примеры изображений в внедренной ЦВЗ существующими методами на экране.

Подход [17], разработанный Якушевым и др. (2021) в Институте Системного Программирования РАН ранее в рамках работы над системой деанонимизации утечек текстовых документов [8], как и метод Gugelmann и др., основан на плавном изменении яркости на экране. Информация кодируется в виде последовательности прямоугольных областей повышенной или пониженной яркости в межстрочные интервалы тестового документа. Незаметность обусловлена тем фактом, что изменение яркости в области с текстом подвержено восприятию значительно меньше в сравнении с одноцветными областями. В каждый межстрочный интервал встраивается 16 бит информации, а для полного встраивания 32-битной последовательности достаточно двух межстрочных интервалов. На рисунке 1.13b приведен пример текстового документа с наложенной при помощи окна-оверлея ЦВЗ. Метод показал высокую устойчивость искажениям, возникающим при фотографировании экрана.

Несмотря на хорошие результаты тестирования, подход плохо подходит для практического применения. Наложение ЦВЗ требует использования алгоритмов поиска текстовых областей на выводимом на экран изображении. Возникает задержка отрисовки ЦВЗ, достигающая нескольких сотен миллисекунд, что может вызвать ощутимый дискомфорт у пользователя устройства. Более того, генерация изображения водяного знака требует значительных вычислительных ресурсов устройства, что особенно критично для рабочих станций невысокой производительности.

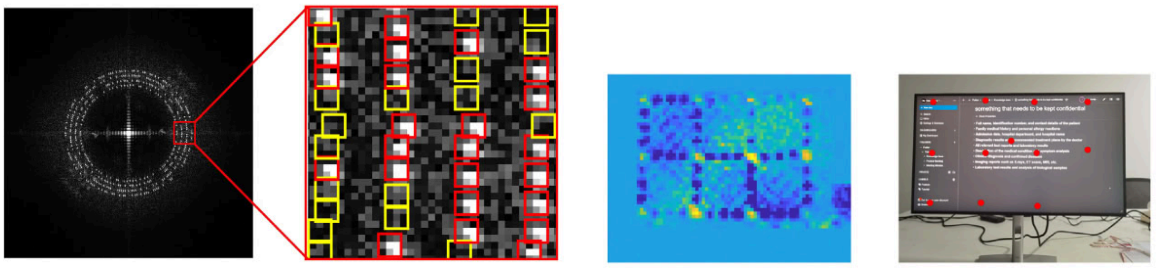
В 2022 году исследователи Ge, Xia, Tong и др. [31] предложили метод внедрения ЦВЗ, генерируемой нейросетевой моделью, в текстовые документы и устойчивый к искажениям возникающим при фотографировании. В процессе обучения задействуется пара нейронных сетей внедрения и извлечения, а также дополнительный слой между ними, имитирующий искажения, возникающие при фотографировании экрана. Используются три функции потерь: первая отвечает за корректность информации, выдаваемой нейросетью извлечения, вторая — за незаметность водяного знака, а третья — не позволяет нейронной сети внедрения изменять области с текстом. Авторы статьи провели сравнение предложенного метода с методами внедрения ЦВЗ в произвольные изображения на экране, примененных к изображениям текстовых документов. Сравнение показало существенное преимущество разработанного подхода. Метод показал высокие результаты по метрикам точности извлечения, но встроенный водяной обладает высокой заметностью (рисунок 1.13c). Несмотря на высокие численные показатели метрик PSNR и SSIM, на изображении со встроенным ЦВЗ хорошо заметны изменения в виде желтых пятен. Также данный метод не позволяет применять подход внедрения ЦВЗ в текстовый документ произвольного формата в режиме реального времени. Нейронная сеть внедрения принимает на вход изображение документа на экране,

современные процессоры общего назначения не достигли достаточной производительности для поддержки достаточно скорости обработки. Метод ориентирован на применение с использованием аппаратных ускорителей вычисления.

Исследователи Грушо, Писковский, Семинихин и др. также работали над проблемой внедрения ЦВЗ в текстовые документы при выводе на экран. В публикациях 2020 [34][14], 2023 [11] и 24 [10] года предлагается встраивать устойчивый к искажениям при фотографировании экрана ЦВЗ на уровне гипервизора. Метод [11] предполагает кодирование информации посредством изменения яркости областей экрана, отвечающих за кодирование определенных символов бинарной последовательности. Предлагается разделить экран на 4 части и генерировать ЦВЗ для каждой четверти с различными параметрами размещения ЦВЗ-символов и плотности кодирования для повышения точности извлечения внедренной информации. Проведенное авторами тестирование показали эффективность метода при считывании закодированной информации из фотографии экрана с выведенным текстовым документом. Оценивалась точность извлечения при фотографировании под различным углом, изменении разрешения, сжатии JPEG, наложении шума и других атаках на ЦВЗ. Однако, исследователи не проводили оценку незаметности ЦВЗ.

В 2024 году опубликована статья [24] Chen и др. о наложении статичного цифрового водяного знака на выводимое на экран изображение при помощи окна-оверлея. Изображение ЦВЗ формируется из изображений двух типов: информационных и паттернов синхронизации. Информационное изображение создается при помощи обратного преобразования Фурье, применяемого к изображению (рисунок 1.14а) с наложенной бинарной последовательностью с кодами обнаружения и исправления ошибок на основе QR-кодов. Изображения синхронизации также формируются при помощи обратного преобразования Фурье и

позволяют выполнять автоматическую коррекцию перспективы фотографии экрана. Метод показал значения метрики BER от 0.03 при съемке 27-дюймового монитора под углом 0° и до 0.1 под углом 45° и расстояния не меньше 40 см. Однако, метод не тестировался на атаках масштабированием к размеру менее 70%, что важно для оценки устойчивости к пересылке фотографий через мессенджеры.



(a) Кодирование информации при помощи IDFT.

(b) Механизм автокоррекции перспективы на основе сетки изображений синхронизации.

Рисунок 1.14. Иллюстрации из статьи [24].

Проведенный анализ существующих подходов показал необходимость разработки метода метода наложения ЦВЗ на экран для защиты на выводимых текстовых документов. Существующие методы не обладают достаточной устойчивостью ЦВЗ к искажениям при фотографировании экрана или слишком заметные для пользователей. Основное требование к методу — он должен быть статическим, то есть, в процессе работы изображение ЦВЗ не должно меняться. Статические ЦВЗ не вызывает дискомфорт у пользователей устройства и расходует минимальные вычислительные ресурсы системы.

1.5 Выводы

В первой главе рассмотрен класс DLP-систем, используемых для противодействия утечкам конфиденциальных документов. DLP-системы способны противостоять различным угрозам информационной

безопасности, однако системы данного класса не способны эффективно защищать *аналоговые* каналы утечек: фотографирование выведенного на экран документа или печать документа с последующей оцифровкой за пределами защищаемого контура при помощи сканера или фотоаппарата.

Проведенный анализ существующих решений, нацеленных на противодействие анонимности утечек конфиденциальных документов, показывает, что четыре из семи рассмотренных продуктов не удовлетворяют требованию о возможности слепого извлечения внедренной в ЦВЗ информации при расследовании утечки. Используемый данными решениями подход требует организации централизованного хранилища документов, что не является возможным для ряда организаций. Согласно доступным материалам, оставшиеся три коммерческих решения позволяют проводить слепое извлечение, однако оценка эффективности и заметности внедряемых ими ЦВЗ затруднена.

Задача внедрения ЦВЗ в документы широко освещена в научных публикациях. Рассмотренные методы внедрения водяных знаков в текстовые документы со слепым извлечением, устойчивые к искажениям, возникающим при печати с последующей оцифровкой сканированием или фотографированием, относятся к структурным методам, использующим текстовую разметку документов или особенности форматирования для кодирования информации. Существующие структурные методы либо обладают недостаточной для внедрения идентификатора сотрудника и устройства емкостью, либо не устойчивы к сжатию изображений текстовых документов при пересылке через мессенджеры. Лингвистические методы демонстрируют высокую устойчивость, но их низкая емкость и риск искажения смысловой нагрузки текста существенно ограничивают их практическое применение. Методы, использующие домен преобразований, подходят в основном для изображений с плавными цветовыми переходами (например, фотографии). Таким образом, задача

разработки метода внедрения ЦВЗ в текстовые документы со слепым извлечением, обладающего устойчивостью к искажениям при печати с последующей оцифровкой сканированием или фотографированием, а также имеющего низкую визуальную заметность, является актуальной.

В литературе представлены методы, позволяющие наносить на документы водяные знаки, устойчивые к искажениям при фотографировании экрана. Динамические методы перестраивают ЦВЗ для адаптации к содержимому экрана, что может потребовать значительных вычислительных ресурсов и вызывать у пользователей дискомфорт из-за частого изменения изображения. Существующие статические методы либо обладают низкой устойчивостью к искажениям, либо не позволяют наносить ЦВЗ поверх документов в режиме реального времени. Защита от анонимных утечек документов на основе подобных методов имеет ограниченную применимость. Таким образом, задача разработки метода наложения ЦВЗ, предполагающего слепое извлечение внедренной информации и обладающего свойствами визуальной незаметности и устойчивости к искажениям, возникающим при фотографировании экрана и сжатии алгоритмами, применяемыми в мессенджерах, остается актуальной.

Глава 2. Система деанонимизации утечек текстовых документов при печати и выводе на экран

Вторая глава содержит пять разделов и описывает архитектуру системы противодействия анонимности при утечках документов. Система состоит из клиентских компонентов, устанавливаемых на *автоматизированные рабочие места* сотрудников (АРМ), и серверной группы компонентов. В разделе 2.1 описаны ключевые механизмы компонентов АРМ, обеспечивающих интеграцию методов внедрения ЦВЗ в документы при печати и выводе на экран.

В разделе 2.2 описан механизм формирования идентификатора сотрудника и устройства, кодируемого в ЦВЗ. Идентификатор представляет собой двоичную последовательность заданной длины. Он формируется при помощи хэш-функции. Идентификатору соответствует набор атрибутов сотрудника и устройства, необходимых для проведения расследования и деанонимизации утечки конфиденциального текстового документа. Также в разделе представлены оценки вероятности коллизии идентификаторов. В разделе 2.3 описан выбранный подход к обнаружению и исправлению ошибок в двоичных последовательностях.

В разделе 2.4 описаны потоки данных между клиентскими и серверными компонентами.

Раздел 2.5 содержит выводы по второй главе.

2.1 Компоненты системы на автоматизированных рабочих местах

Система деанонимизации утечек текстовых документов имеет клиент-серверную архитектуру. Клиентские компоненты системы устанавливаются на АРМ сотрудников организации и содержат реализацию методов внедрения ЦВЗ в текстовые документы при печати и

выводе на экран, а также средства интеграции методов внедрения ЦВЗ в *операционную систему* (ОС). Предполагается, что сотрудники организации не имеют прав администратора АРМ. Следовательно сотрудники не смогут повлиять на работу служб и сервисов запущенным администратором. Стоит отметить, что данной политики придерживается большинство крупных организаций в отношении сотрудников, в рабочие обязанности которых не входит обслуживание АРМ.

Компоненты внедрения ЦВЗ в текстовые документа разделяются на две категории: компоненты внедрения ЦВЗ при печати и компоненты наложения ЦВЗ на экран монитора. Первые обеспечивают защиту текстовых документов от анонимных утечек фотографий или сканов распечатанных конфиденциальных документов, вторые защищают от анонимных утечек фотографий экрана с выведенным конфиденциальным документом. Разработанные методы внедрения ЦВЗ подробно описаны в главах 3 и 4.

Методы внедрения ЦВЗ при печати ориентированы на работу с изображениями текстовых документов. Внедрение ЦВЗ выполняется бесшовно для пользователя и приложения, в которым он работает. При установке системы на рабочую станцию под управлением ОС семейства Microsoft Windows создается и конфигурируется виртуальный принтер, осуществляющий внедрение ЦВЗ в документ и последующее перенаправление копии документа с ЦВЗ на физический принтер. При печати документа из приложения работы с документами, например, Microsoft Word или Adobe Reader, пользователь должен выбрать виртуальный принтер в качестве устройства печати. Соответственно, при установке системы деанонимизации утечек на рабочую станцию должны применяться политики, запрещающие печать через иные устройства помимо виртуального принтера.

Виртуальный XPS-принтер (XML Paper Specification) для операционной системы Windows реализован на основе *драйвера универсального принтера Microsoft* (Unidrv) [64], позволяющего гибко управлять процессом печати и работающего в пространстве пользователя, а не ядра. Документы при печати конвертируются в универсальное представление XPS. Драйвер виртуального XPS-принтера представляет собой конвейер из одного или нескольких XPS-фильтров, работающих независимо друг от друга. Компонент, обеспечивающий внедрение ЦВЗ в текстовый документ, реализован как XPS-фильтр и интегрирован как один из этапов обработки документа, отправленного на печать в виртуальный XPS-принтер. Поскольку алгоритмы внедрения ЦВЗ ориентированы на работу с изображениями, в рамках XPS-фильтра при необходимости выполняется растеризация страниц документа.

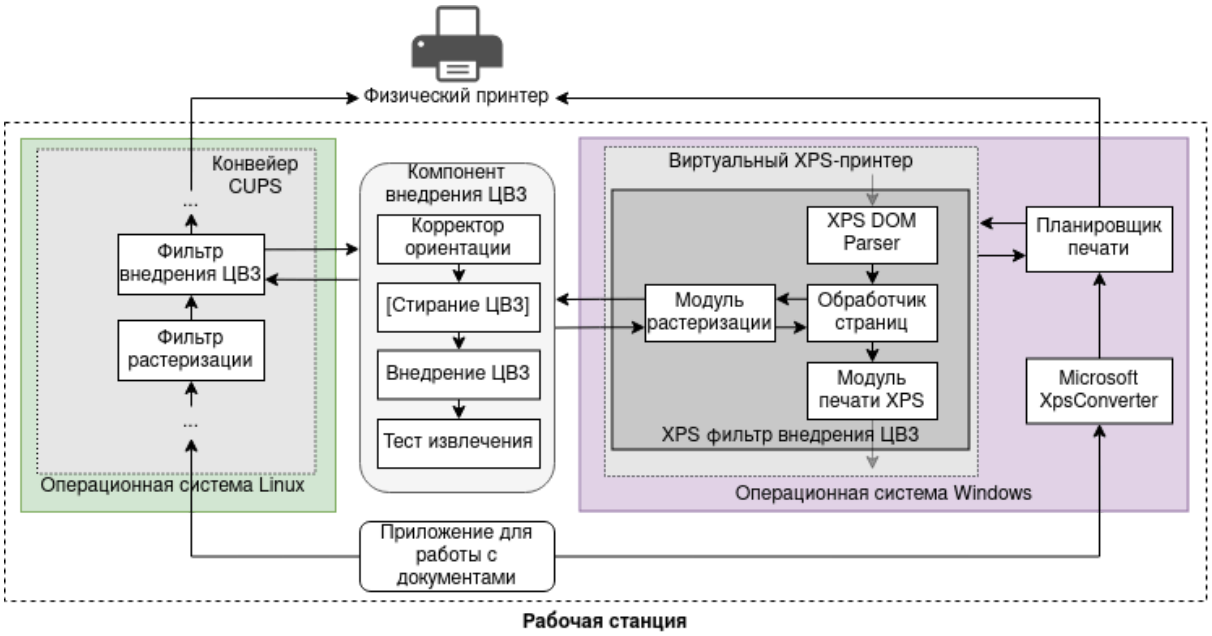


Рисунок 2.1. Схема внедрения ЦВЗ в текстовые документы при печати.

В ОС семейства Linux процесс внедрения ЦВЗ реализован через подсистему CUPS (Common UNIX Printing System), используемую в UNIX-подобных ОС для управления печатью и работы с принтерами. В рамках CUPS документ проходит через цепочку CUPS-фильтров, выполняющих преобразование форматов документов в тот вид, который

может быть распознан заданным физическим принтером. Цепочки фильтров задают карту преобразований форматов документов при печати. При установке компонентов системы в каждый путь карты преобразований добавляются фильтры предобработки и внедрения ЦВЗ в документ.

Виртуальный принтер и CUPS-фильтр выполняют функции: растеризация и разбор многостраничного документа на отдельные страницы, коррекция ориентации страницы, внедрение ЦВЗ в каждую страницу документа. На печать отправляется страница с внедренным ЦВЗ. Но если работа алгоритма внедрения ЦВЗ завершилась с ошибкой, печатается исходная (без ЦВЗ) страница. Стратегия полного запрета печати документов без водяного знака не применяется, поскольку она может создавать значительные сложности для пользователей, а также сфокусировать их внимание на различиях между распечатанными документами и документами, отправленными на печать. При печати документа с ранее внедренным ЦВЗ водяной знак заменяется на новый, содержащий данные о пользователе, инициировавшем текущую печать.

Механизм наложения цифрового водяного знака на документы при выводе на экран реализован при помощи окна-оверлея. Водяной знак отображается поверх всех остальных окон и является статическим. Накладываемое изображение генерируется при запуске программы и не изменяется на протяжении сеанса работы пользователя за исключением таких событий, как изменение разрешения экрана, подключение дополнительного монитора и других подобных. Программа внедрения ЦВЗ “на экран” получает из ОС информацию о состоянии графического интерфейса, положении и содержании окон графических приложений и активных дисплеях. Для того, чтобы окно-оверлей постоянно поддерживалось поверх всех остальных окон, на АРМ под управлением ОС семейства Microsoft Windows для него выставляется свойство

`Topmost` [62], а для АРМ под управлением ОС семейства Linux применяется системный вызов `set_keep_above` [32].

Композитные менеджеры ОС позволяют добавлять на выводимое на экран изображение различные визуальные эффекты, в том числе, эффект прозрачности. Окно-оверлей отображает ЦВЗ с заданным коэффициентом непрозрачности α. Согласно правилам сложения изображений [43], обладающих свойством частичной прозрачности, значения пикселей изображения с ЦВЗ I'^{c}_{s}, выводимого на экран, вычисляются по формуле:

$$I'^{c}_{s} = (1 - \alpha) \cdot I^{c}_{s}(x, y) + \alpha \cdot I^{c}_{o}(x, y),$$

где $c \in \{R, G, B\}$ – цветовые каналы, I_s – выводимое изображение на экран, а I_o – изображение ЦВЗ.

2.2. Идентификатор сотрудника и устройства

Представленные в главах 3 и 4 методы внедрения ЦВЗ предназначены для встраивания коротких битовых последовательностей, порядка нескольких десятков бит. Увеличение емкости водяного знака приводит к снижению его устойчивости и уменьшает количество подходящих для встраивания документов.

Водяной знак, встроенный в документ, должен содержать информацию, необходимую для идентификации пользователя и устройства для определения источника утечки. Однако емкости в несколько десятков бит, как правило, недостаточно для хранения данных, представленных в виде текстовой строки. Поэтому был выбран подход, при котором внедряемый в документ ЦВЗ кодирует *идентификатор*. Данный идентификатор ассоциирован с набором атрибутов, необходимых для расследования утечки. Идентификатор и набор атрибутов отправляются с устройства в базу данных на сервере. В случае утечки для определения

виновника потребуется извлечь внедренный ЦВЗ, получить идентификатор, найти его в базе данных.

Рисунок 2.2. Пример серийного номера жесткого диска от производителя Western Digital.

Во многих организациях используются системы доменных учетных записей, что позволяет сотрудникам работать на различных АРМ с использованием только одной учетной записи. В этой связи (для получения информации об АРМ, где произошла утечка) видится разумным включение в перечень передаваемых на сервер атрибутов как атрибутов учетной записи, так и атрибутов рабочей станции. В число атрибутов рабочей станции должны входить уникальные идентификаторы. Изначально может показаться, что инвентарный номер рабочей станции является подходящим атрибутом, однако инвентарные данные часто не актуализируются. Кроме того, для каждой рабочей станции потребуется ручной ввод инвентарного номера, что усложняет автоматическое развертывание системы в организации. Более подходящей альтернативой является серийный номер ПЗУ, поскольку его можно автоматически считать, что упрощает

внедрение. Производители ПЗУ, как правило, используют собственные системы генерации серийных номеров и поддерживают их уникальность, что делает этот атрибут достаточно надежным. Учетная запись сотрудника может характеризоваться следующими атрибутами:

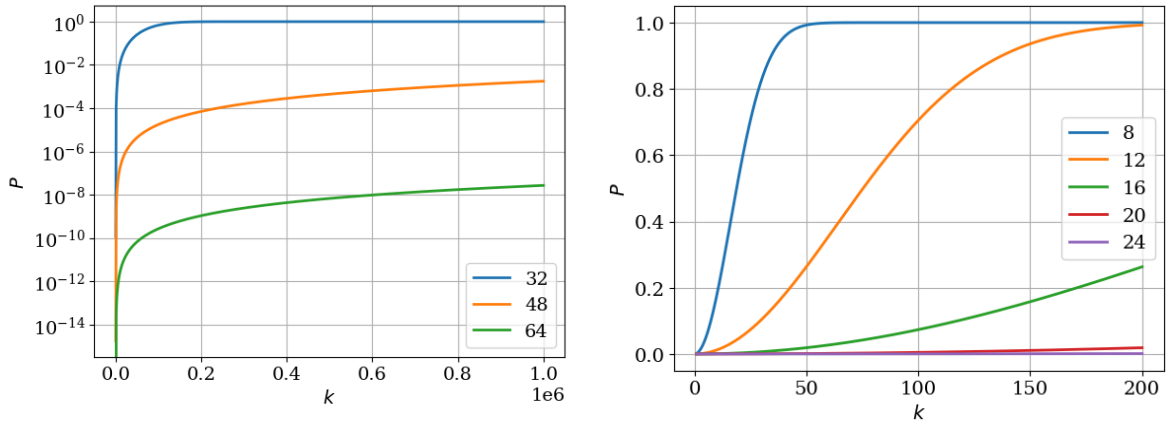
- домен учетной записи;
- имя учетной записи сотрудника.

Существуют различные способы преобразования набора атрибутов пользователя в уникальный идентификатор — битовую последовательность заданной длины. Хэш-функции позволяют преобразовывать строку произвольной длины в битовую последовательность фиксированной длины. В то же время, чем короче длина хэш-кода, тем выше вероятность возникновения коллизий. Коллизия в контексте хэш-функций — это ситуация, когда для двух различных входов получается одинаковый результат, т.е. если хэш-функция $H(x)$ отображает два разных входа x_1 и x_2 в одно и то же значение $H(x_1) = H(x_2)$.

Вероятность коллизии для хэш-функции размерности m бит можно оценить с использованием *парадокса дней рождения*. Предположим, что имеется k различных входов, и хэш-функция отображает аргументы в 2^m возможных значений. Формула для вероятности хотя бы одной коллизии среди k различных элементов:

$$P = \frac{2^m!}{2^{mk}(2^m - k)!}$$

При значениях k не превышающих порядок $2^{m/2}$ допускается использовать аппроксимацию $P \approx 1 - e^{-k^2/2^{m+1}}$.



(a) (b)

Рисунок 2.3. Оценка вероятности коллизии P при значениях больших (a) и небольших (b) значениях k и m (значения отображены в легенде).

Для того, чтобы оценить значение k, при котором вероятность коллизии p превышает некоторое пороговое значение δ, преобразуем уравнение аппроксимации вероятности коллизии:

$$\delta = 1 - e^{-k_m^2/2^{m+1}}$$

$$-\frac{k_m^2}{2^{m+1}} = ln(1 - \delta)$$

$$k_m = \sqrt{-2^{m+1} \cdot ln(1 - \delta)}$$

В результате можно дать оценку значений k_m при заданных пороговой вероятности коллизии δ – расчеты представлены в таблице 2.1.

Таблица 2.1. Значения k_m, при которых вероятность коллизии превышает заданный порог δ.

m	$k_m \mid \delta = 1\%$	$k_m \mid \delta = 5\%$	$k_m \mid \delta = 10\%$	$k_m \mid \delta = 50\%$
16	36.295	81.995	117.515	301.417
24	580.718	1311.914	1880.243	4822.671
32	9291.487	20990.618	30083.882	77162.743
48	$2.378 \cdot 10^6$	$5.373 \cdot 10^6$	$7.701 \cdot 10^6$	$1.975 \cdot 10^7$
64	$6.089 \cdot 10^8$	$1.375 \cdot 10^9$	$1.971 \cdot 10^9$	$5.056 \cdot 10^9$

Коллизия пользовательских идентификаторов может привести к тому, что при расследовании утечки у сотрудника службы безопасности окажется несколько подозреваемых, что усложняет процесс. Важно минимизировать вероятность коллизий. Число сотрудников в организации, где используется система, напрямую влияет на минимально допустимую длину идентификатора. Для малых организаций достаточно идентификатора длиной 16 бит, для средних — 24 бит, а для крупных потребуется 32-битный идентификатор. Если в организации более 10 000 сотрудников, вероятность коллизии для 32-битного хэш-кода превышает 1%, поэтому рекомендуется использовать хэш-коды большей длины.

Таблица 2.2. Расчеты значений максимального числа сотрудников организации k_{max}.

n	m	$k_{max}\|\delta = 1\%$	$k_{max}\|\delta = 5\%$	$k_{max}\|\delta = 10\%$	$k_{max}\|\delta = 50\%$
24	16	9291.487	20990.618	30083.882	77162.743
32	16	$2.378 \cdot 10^6$	$5.373 \cdot 10^6$	$7.701 \cdot 10^6$	$1.975 \cdot 10^7$
32	24	$1.486 \cdot 10^5$	$3.358 \cdot 10^5$	$4.813 \cdot 10^5$	$1.234 \cdot 10^5$
48	24	$9.742 \cdot 10^{10}$	$2.201 \cdot 10^{10}$	$3.154 \cdot 10^{10}$	$8.091 \cdot 10^{10}$
48	32	$6.089 \cdot 10^8$	$1.375 \cdot 10^9$	$1.971 \cdot 10^9$	$5.056 \cdot 10^9$
64	32	$3.990 \cdot 10^{13}$	$9.015 \cdot 10^{13}$	$1.292 \cdot 10^{14}$	$3.314 \cdot 10^{14}$

Часто в крупных организациях существует структура подразделений, что позволяет распределить сотрудников по меньшим организационным единицам. Каждому подразделению можно присвоить уникальный номер и разделить емкость n бит идентификатора на две части: хэш-код атрибутов учетной записи сотрудника длиной m бит и номер департамента длиной $n - m$ бит. Максимальное количество департаментов в организации при этом составит 2^{n-m}. Таким образом, максимальное число сотрудников в организации при наибольшем числе сотрудников в департаменте k_m можно определить как:

$$k_{max} = 2^{n-m} \cdot k_m \mid p = \delta.$$

Крупнейшая организация в Российской Федерации, ОАО «РЖД», имеет численность персонала в 701,2 тыс. человек на 2022 год [30]. ЦВЗ емкостью 32 бита достаточно для подобных по численности организаций с разделением на департаменты (см. Таблицу 2.2).

2.3 Обнаружение и исправление ошибок при извлечении идентификатора сотрудника и устройства

При декодировании внедренной в ЦВЗ информации могут возникать ошибки. Основной причиной ошибок являются искажения, возникающие при печати, фотографировании, сканировании и сжатии изображения документа. Для надежного извлечения внедренной в ЦВЗ информации требуются механизмы обнаружения и исправления ошибок.

2.3.1 Подходы к обнаружению и исправлению ошибок в битовых последовательностях

Применение технологии ЦВЗ предполагает наличие двух этапов: *внедрение* двоичной последовательности $m, m_i \in \{0,1\}, i \in \{1, \cdots, M\}$ в изображение документа D и *извлечение* двоичной последовательности $m', m_i' \in \{0,1\}, i \in \{1, \cdots, M'\}$ из искаженного документа D_w' с внедренным ЦВЗ. Извлечение считается корректным, если $m = m'$, требуется обрабатывать ситуации некорректного извлечения, когда $m \neq m'$.

Метод вычисления ***бита четности*** [16] является одним из самых простых для обнаружения ошибок. К исходной двоичной последовательности добавляется один бит (бит четности), значение которого равно 1, если число единичных битов в исходной последовательности нечетное, и 0 в противном случае. К недостаткам метода можно отнести слабую стойкость к ошибкам инвертирования битов

— очевидно, метод не чувствителен к четному количеству ошибок инвертирования битов в m'.

$$\widehat{m} = m \vee m_{PB},\ m_{PB} = m_1 \oplus \quad \cdots \quad \oplus m_M$$

При использовании ***многомерных кодов*** [16] данные организуются в двумерную или даже многомерную матрицу, для каждого измерения которой вычисляются контрольные биты четности. Это позволяет не только обнаруживать, но и исправлять ошибки. Положение инвертированных контрольных битов позволяет вычислить местоположение инвертированных битов данных в матрице. Бинарная последовательность m может быть представлена в виде матрицы:

$$m = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1q} \\ m_{21} & m_{22} & \dots & m_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ m_{p1} & m_{p2} & \dots & m_{pq} \end{bmatrix},\ M = pq$$

Бит четности для строки i определяется как

$$b_{i(q+1)} = m_{i1} \oplus m_{i2} \oplus \cdots \oplus m_{iq}$$

Бит четности для столбца j определяется как

$$b_{(p+1)j} = m_{1j} \oplus m_{2j} \oplus \cdots \oplus m_{pj}$$

$$\hat{m} = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1q} & b_{1(q+1)} \\ m_{21} & m_{22} & \dots & m_{2q} & b_{2(q+1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ m_{p1} & m_{p2} & \dots & m_{pq} & b_{p(q+1)} \\ b_{(p+1)1} & b_{(p+1)2} & \dots & b_{(p+1)q} & b_{(p+1)(q+1)} \end{bmatrix}$$

Проверка извлеченной битовой последовательности на наличие ошибок выполняется посредством расчета битов четности b_{ij} для каждого столбца и строки. Если значения всех битов четности совпадают с вычисленными, ошибок нет. Пересечение строки и столбца с неверными битами четности указывает на позицию ошибки. Многомерные коды

позволяют исправлять одиночные ошибки, но при большем количестве ошибок эффективность подхода падает. Также для реализации многомерных кодов требуется большее количество битов с сравнении с одномерными подходами.

Циклический избыточный код (CRC, Cyclic Redundancy Check) [16] — это метод обнаружения ошибок в данных, основанный на делении бинарной последовательности на некоторый заданный полином и использовании остатка от этого деления в качестве контрольной суммы. Кодируемая последовательность m представляется в виде полинома $D(x)$ степени $M - 1$ с коэффициентами из множества $\{0, 1\}$. Заранее определяется генеративный полином (полином делитель) $G(x)$ степени n с коэффициентами из $\{0, 1\}$. Контрольная сумма CRC вычисляется из:

$$D(x) \times x^n = Q(x) \times G(x) + R(x)$$

Здесь:

- $Q(x)$ — частное;
- $R(x)$ — полином остатка, имеющий степень не выше $n - 1$;
- Операция $D(x) \times x^n$ эквивалентна добавлению n нулей к исходной последовательности.

Контрольная сумма представляется полиномом $R(x)$ и добавляется к исходным данным: $T(x) = D(x) \times x^n + R(x)$. Для обнаружения ошибок полученная последовательность $T(x)$ делится на полином $G(x)$, и если остаток равен нулю, то ошибки отсутствуют.

$$T(x) \, mod \, G(x) = 0$$

CRC-32 — один из наиболее широко используемых вариантов CRC для проверки целостности данных. Он применяется в различных сетевых протоколах, в частности Ethernet и Wi-Fi, для проверки целостности пакетов данных. Форматы сжатия файлов, например ZIP и RAR, также используют CRC-32 для проверки целостности архивов. CRC-32

генерирует контрольную сумму длиной 32 бита, что означает, что остаток от деления данных на генераторный полином представлен в виде 32-битного числа. Этот метод позволяет обнаруживать большинство одиночных и двойных ошибок, а также многие виды групповых ошибок. В то же время, CRC-32 может только обнаружить ошибку, но не исправить её.

Код Хэмминга [35] — это один из наиболее известных методов кодирования данных для обнаружения и исправления ошибок. Код Хэмминга является систематическим блоковым линейным кодом, который добавляет r битов четности к k битам данных таким образом, что может обнаруживать и исправлять одиночные ошибки. Для того чтобы код Хэмминга мог обнаружить и исправить одиночную ошибку, количество контрольных бит r должно удовлетворять неравенству:

$$2^r \geq k + r + 1$$

Кодовое слово длиной $n = k + r$ состоит из k информационных битов и r битов четности, расположенных в определённых позициях (обычно на степенях двойки). Каждый бит четности b_i определяется как линейная комбинация (по модулю 2) определенных битов данных. Бит четности b_i контролирует множество S_i битов данных, чьи номера в бинарной записи содержат единицу в i-м разряде:

$$b_i = \bigoplus_{j \in S_i} m_j$$

Для обнаружения ошибок вычисляются и сравниваются биты четности. Если значения не совпадают, то вычисляется синдром S. Ненулевой синдром указывает на наличие ошибки, а его значение определяет позицию ошибочного бита:

$$S = \sum_{i=1}^{r} b_i \cdot 2^{i-1}$$

Код Рида-Соломона (Reed-Solomon code) [47] — это линейный блочный код, широко используемый для обнаружения и исправления ошибок в данных. Код Рида-Соломона оперирует над символами, представляющими собой группы битов (например, байты), и каждая группа обрабатывается как единое целое. Код Рида-Соломона работает в конечном поле $GF(q)$, $q = 2^p$ и p — целое положительное число, определяющее количество битов в символе. Код определяется параметрами $(n,\ k)$ где:

- n — длина кодового слова, состоящего из информационных символов (исходные данные) и проверочных символов (избыточные символы);
- k — количество информационных символов.

Для формирования кодовых слов используется генераторный многочлен $g(x)$, имеющий степень $n - k$ и определяемый как:

$$g(x) = (x - \alpha_1)(x - \alpha_2) \cdots (x - \alpha_{n-k})$$

где α_i, $i \in [1, n - k]$ это $(n - k)$ корней полинома кодового слова, выбранных из конечного поля $GF(q)$. Кодовое слово определяется как:

$$c(x) = m(x) \cdot x^{n-k} + r(x)$$

где $m(x)$ — многочлен степени $k - 1$, формируемый на основе битовой последовательности m, а $r(x)$ — многочлен степени не более $n - k - 1$, являющийся остатком от деления $m(x) \cdot x^{n-k}$ на генераторный многочлен $g(x)$. Таким образом, $r(x)$ определяется как:

$$r(x) = (m(x) \cdot x^{n-k})\ mod\ g(x)$$

Код Рида-Соломона позволяет исправлять до $t = (n - k)/2$ ошибочных символов и обнаруживать до $2t$ ошибочных символов в кодовом слове. Обнаружение и исправление ошибок выполняется при помощи синдромов $S_i = c(\alpha_i),\ i = 1, 2, \ldots, n - k$, представляющих собой значения многочлена,

вычисленного в точках, определенных корнями генеративного многочлена. Если один или несколько синдромов ненулевые, это указывает на наличие ошибок. С помощью алгоритма декодирования (например, алгоритма Берлекэмпа — Мэсси [16]) можно найти и исправить ошибочные символы.

БЧХ-коды (Коды Боуза-Чоудхури-Хоквингема) [21] — это класс циклических кодов, применяемых для защиты информации от ошибок. Код Рида-Соломона является частным случаем БЧХ-кодов. Код работает в конечном поле $GF(q)$, $q = 2^p$, где p — целое положительное число, элементы поля представлены как мночлены степени $p - 1$ с коэффициентами из $GF(2)$. Код определяется параметрами $(n,\ k,\ d)$, где:

- $n = \frac{q-1}{s}$ — длина кодового слова (в символах), при $s = 1$ код называется примитивным;
- k — количество информационных символов, $r = n - k$ — количество проверочных символов;
- d — минимальное кодовое расстояние, то есть минимальное количество позиций, в которых различаются любые два кодовых слова ($d < n$).

Данный код позволяет исправлять до $t = \frac{d-1}{2}$ ошибок в кодовом слове. Пусть α — примитивный элемент поля $GF(q)$. Генераторный полином $g(x)$ БЧХ-кода имеет степень $n - k$ и является наименьшим общим кратным многочленов $p_{\alpha^i}(x)$, где $p_{\alpha^i}(x)$ — минимальный многочлен элемента α^i, $1 \leq i < d$.

$$g(x) = LCM(p_{\alpha^1}(x), p_{\alpha^2}(x), \cdots), p_{\alpha^{d-1}}(x))$$

Кодовое слово $c(x)$ представляет собой многочлен степени $n - 1$, кратный генераторному многочлену:

$$c(x) = m(x) \cdot g(x)$$

где $m(x)$ — многочлен данных степени не более $k-1$. Для декодирования и исправления ошибок используются синдромы, которые вычисляются на основе значений многочлена $c(x)$ в корнях генераторного многочлена.

Пусть $r(x)$ — кодовое слово, которое может содержать ошибки. Задача декодера состоит в том, чтобы обнаружить ошибки в $r(x)$ и исправить их, чтобы получить исходное переданное кодовое слово $c(x)$. Синдромы вычисляются как остаток от подстановки корней генераторного многочлена в полученное слово:

$$S_i = r(\alpha^i),\ i = 1, 2, \cdots, 2t$$

Если все синдромы равны нулю, то ошибок нет. В противном случае существует как минимум одна ошибка.

Корни полинома локатора ошибок $\sigma(x)$ указывают на позиции ошибок. Полином имеет вид:

$$\sigma(x) = 1 + \sigma_1 x + \sigma_2 x^2 + \quad \cdots \quad + \sigma_t x^t$$

Коэффициенты полинома находятся решением системы линейных уравнений, полученных на основе синдромов:

$$S_i + \sigma_1 S_{i-1} + \sigma_2 S_{i-2} + \quad \cdots \quad + \sigma_t S_{i-t} = 0,\ i = t+1, t+2, \cdots, 2t$$

Для нахождения корней необходимо найти такие значения $x = \alpha^j$, при которых полином $\sigma(x) = 0$. Если α^j — корень полинома, то позиция ошибки находится на j-й позиции кодового слова.

Амплитуды ошибок, то есть величины, которые необходимо добавить к ошибочным символам, чтобы исправить их, вычисляются с помощью полинома локаторов значений ошибок $\Omega(x)$. Этот полином связан с синдромами и полиномом локаторов ошибок:

$$\Omega(x) = S(x) \cdot \sigma(x)\ mod\ x^{2t}$$

Амплитуды ошибок вычисляются через резольвенты, используя полином $\sigma(x)$ и его производную. После нахождения позиций ошибок и их амплитуд производится корректировка соответствующих символов в кодовом слове. Если позиция j была определена как ошибочная, и амплитуда ошибки равна e_j, то корректировка производится следующим образом:

$$r_j' = r_j - e_j$$

где r_j — символ в позиции j в полученном кодовом слове, а r_j' — исправленный символ.

БЧХ-коды являются наиболее подходящим решением для обнаружения и исправления ошибок в коротких битовых последовательностях. Они обеспечивают высокий уровень надежности, позволяя исправлять несколько ошибок одновременно. В отличие от многомерных битов четности и кода Хэмминга, которые обладают ограниченными возможностями по исправлению ошибок, БЧХ-коды более гибкие и мощные. Несмотря на свою эффективность, коды Рида-Соломона ориентированы на работу с большими блоками данных и избыточны для коротких последовательностей. CRC, хотя и прекрасно подходит для обнаружения ошибок, не может их исправлять, что снижает его применимость в задачах, требующих восстановления данных. Таким образом, БЧХ-коды являются оптимальным выбором для обнаружения и исправления ошибок при работе с битовыми последовательностями, в частности в контексте извлечения информации из ЦВЗ.

2.3.2 Анализ применимости БЧХ-кода для обнаружения и исправления ошибок при извлечении идентификатора сотрудника и устройства

БЧХ-код является циклическим кодом, позволяющим исправлять до $t \leq \frac{n-k}{p}$ ошибок в кодовом слове длины n, состоящем из

информационного слова длиной k и проверочных символов длиной $n - k$. На основе данных параметров подбирается генераторный полином $g(x)$. БЧХ-код может быть двух видов:

- *несистематический* – кодовый блок требует дешифровки для прочтения информационного слова;
- *систематический* – кодовое слово формируется в результате конкатенации информационного слова и проверочных символов.

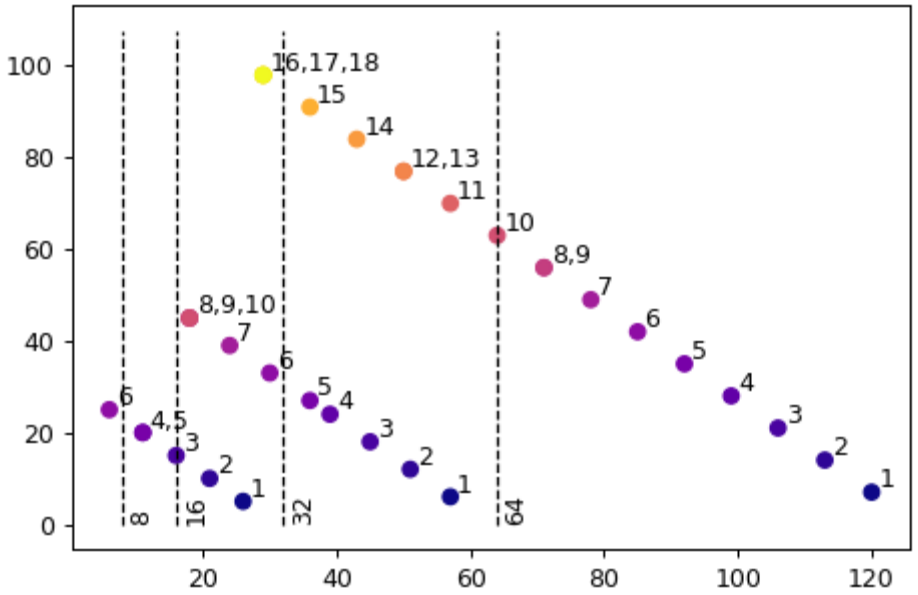


Рисунок 2.4. Длина проверочного кода и кодового слова в зависимости от параметра t. На горизонтальной оси отмечена длина кодового слова, на вертикальной – длина проверочного кода, над точками указано значение параметра t.

В условиях ограниченной емкости контейнера цифрового водяного знака большую важность имеет значение $\frac{n-k}{k}$ — отношение длины проверочного слова к информационному (меньше — лучше). При фиксированном значении t более эффективен БЧХ-код с большими значениями параметра p (и следовательно n). При информационном слове фиксированной длины, состоящем из одного или более октетов, оптимальным является БЧХ-код с параметром p, имеющим минимальное и достаточное значение. Для кодового слова длиной 16 бит оптимально

использование БЧХ-кода с параметром $p = 5$, для 32 бит $p = 6$ и для 64 бит $p = 7$. Расчеты представлены в таблице ниже.

Таблица 2.3. Параметры БЧХ-кода для коротких кодовых слов.

(p, n, k)	t	$\frac{n-k}{k}$	$\frac{n-k}{16}$	$\frac{n-k}{32}$	$\frac{n-k}{64}$
(5, 31, 26)	1	0.192	0.312	-	-
(5, 31, 21)	2	0.476	0.625	-	-
(5, 31, 16)	3	0.937	0.937	-	-
(5, 31, 11)	5	1.818	1.25	-	-
(6, 63, 57)	1	0.105	0.375	0.187	-
(6, 63, 51)	2	0.235	0.75	0.375	-
(6, 63, 45)	3	0.4	1.125	0.562	-
(6, 63, 39)	4	0.615	1.5	0.75	-
(6, 63, 36)	5	0.75	1.687	0.844	-
(6, 63, 30)	6	1.1	2.062	1.031	-
(7, 127, 120)	1	0.058	0.437	0.219	0.109
(7, 127, 113)	2	0.116	0.875	0.437	0.219
(7, 127, 106)	3	0.198	1.312	0.656	0.328
(7, 127, 99)	4	0.264	1.75	0.875	0.437
(7, 127, 92)	5	0.330	2.187	1.093	0.547
(7, 127, 85)	6	0.494	2.625	1.312	0.656
(7, 127, 78)	7	0.628	3.062	1.531	0.766

В кодовом слове $r(x)$ могут возникать ошибки нескольких типов:

- ошибка *инверсии* — значение бита изменено на противоположное в некоторой позиции кодового слова;
- *пропуск* бита — значение бита в некоторой позиции кодового слова неизвестно;
- ошибки *синхронизации* — изменение длины кодового слова:
    - *вставка* — добавление бита в некоторую позицию кодового слова;

- *удаление* — удаление бита в некоторой позиции кодового слова.

БЧХ-код позволяет исправлять до t и гарантированно обнаруживать до $2t$ ошибок инверсии, при большем числе ошибок обнаружение не гарантируется. Таким образом, при помощи БЧХ-кода можно исправлять большее чем t число ошибок с заданной вероятностью при помощи алгоритмов перебора. Стоит отметить, что возможны варианты кодового слова $r(x) \neq c(x)$ и корректным БЧХ-кодом, обозначим такие варианты как *коллизии*.

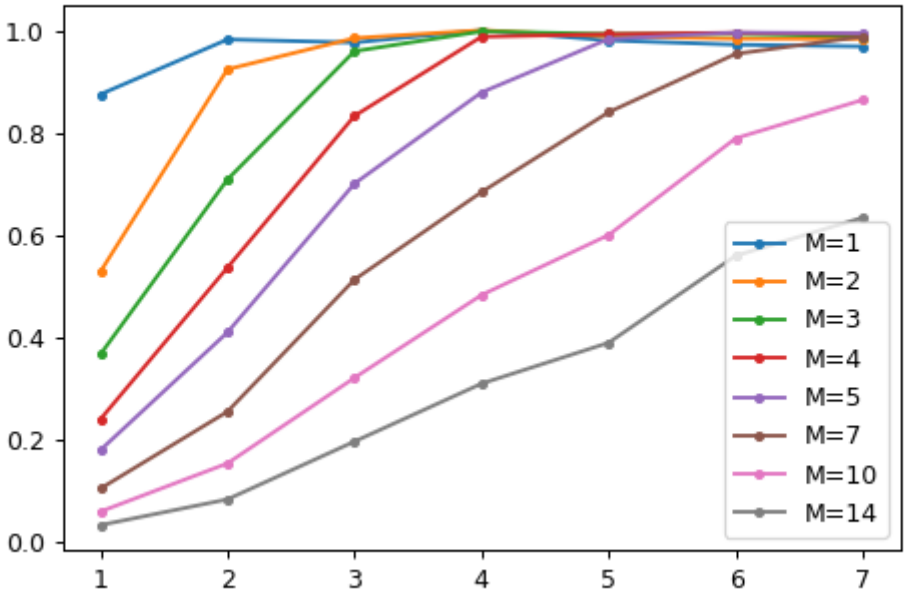


Рисунок 2.5. Зависимость доли верно исправленных $r(x)$ от параметра t. На горизонтальной оси: максимальное количество корректируемых БЧХ-кодом ошибок (параметр t), на вертикальной: доля успешно скорректированных кодовых слов.

Если в кодовой последовательности могут возникать ошибки пропуска и синхронизации, то для исправления ошибок предлагается выполнить перебор возможных значений на месте пропущенных бит до тех пор, пока не будет найдено кодовое слово $r(x)$ с числом ошибок меньше t. Для оценки частоты появления коллизий при внесении ошибок в кодовом слове с фиксированной длиной информационного слова в 32 бита и параметром $1 \leq t \leq 7$ был проведен эксперимент. Для каждого

значения параметра было сгенерировано по 50 случайных битовых последовательностей m, и в каждый вариант кодового слова вносилось по 100 ошибок обоих типов в случайном количестве с математическим ожиданием M. Результаты эксперимента представлены на рисунках 2.5 и 2.6. При общем числе ошибок не превышающем t исходное кодовое слово можно получить с высокой вероятностью $> 90\%$, исправление большего числа ошибок инверсии маловероятно. Однако, при $t \geq 4$ возможно исправление до $2t$ ошибок пропуска с высокой вероятностью $> 90\%$.

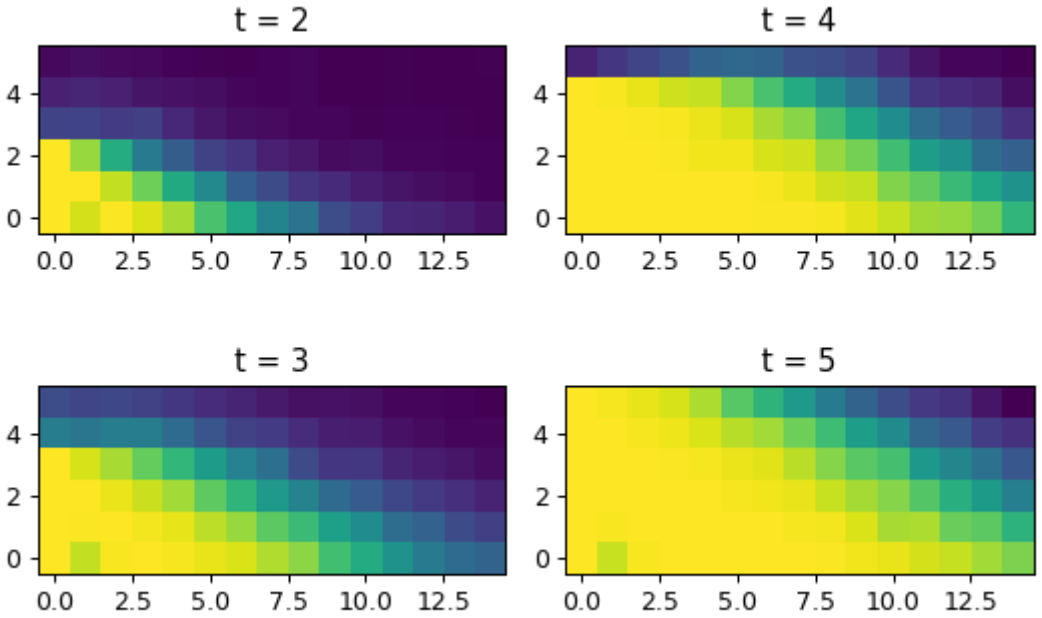


Рисунок 2.6. Тепловая карта доли корректируемых меток для БЧХ-кодов, исправляющих максимум t ошибок. На вертикальной оси: количество ошибок инверсии, на горизонтальной: количество ошибок пропуска.

При наличии в кодовой последовательности ошибок синхронизации эффективность БЧХ-кодов существенно снижается. Коды Варшамова-Тененгольца [1] позволяют исправлять ошибки синхронизации, однако только единичные. В рамках диссертационной работы предложен алгоритм поиска исходной последовательности $c(x)$ при ошибках синхронизации с использованием БЧХ-кодов. Алгоритм 2.1 основан на переборе возможных положений ошибок синхронизации в кодовом слове и поиске набора ошибок синхронизации, имеющих

минимальное количество ошибок инверсии. Алгоритм имеет высокую алгоритмическую сложность и неэффективен при большой длине последовательности $\overline{M}$, также при переборе комбинаций с большим числом ошибок синхронизации растет вероятность коллизии БЧХ-кода. Результатом работы алгоритма является множество бинарных последовательностей V, в которой может находиться исходная битовая последовательность $c(x)$.

Алгоритм 2.1. Поиск исходного кодового слова по кодовому слову с ошибками синхронизации.

function $MutateSeq(M, I, P)$

Input: Битовая последовательность $M\,|M_i \in \{0,1\}$, $1 \le i \le \overline{M}$; кортеж индексов $I\,|1 \le I_i \le \overline{M}$ длиной E_{syn}; кортеж изменений $P\,|P_i \in \{D_1, D_0, I_0, I_1\}$ длиной E_{syn}

Output: Измененная битовая последовательность $M'|$ $M_i' \in \{0,1\},\ \ 1 \le i \le \overline{M}$

$M' \leftarrow M$

for each i **in** I **do**

  for each p **in** P **do**

    switch p **of**

      case Del_1: $M' \leftarrow M \setminus \{M_i\} \cup \{1\}$

      case Del_0: $M' \leftarrow M \setminus \{M_i\} \cup \{0\}$

      case Ins_0: $M' \leftarrow M_1 M_2 \cdots M_{i-1} 0 M_i \cdots M_{\overline{M}-1}$

      case Ins_1: $M' \leftarrow M_1 M_2 \cdots M_{i-1} 1 M_i \cdots M_{\overline{M}-1}$

return M'

function $BruteforceBCH(M,\ max\, E_{syn})$

Input: Битовая последовательность $M \mid \quad M_i \in \{0,1\}$, $1 \le i \le \overline{M}$; максимальное число исправляемых ошибок синхронизации $max\, E_{syn}$

Output: *Множество битовых последовательностей* $V \mid V_k = M'$, $M_i' \in \{0,1\}$, $1 \le i \le \overline{M}$ с bf_{min} исправленных бит при помощи БЧХ-кода

$Perm \leftarrow \{Del_1, Del_0, Ins_0, Ins_1\}$

$S \leftarrow \varnothing,\ V \leftarrow \varnothing,\ bf_{min} \leftarrow t$

for each E_{syn} **in** $\{1, 2, \cdots, max\, E_{syn}\}$ **do**

  for each I **in** $I^{E_{syn}} = I_1 \times \cdots \times I_{E_{syn}} \mid I_i = \{1, \cdots, \overline{M}\}$ **do**

    for each P **in** $P^{E_{syn}} = P_1 \times \cdots \times P_{E_{syn}} \mid P_i = Perm$ **do**

      $M' \leftarrow MutateSeq(M, I, P)$

      if $M' \in S$ **then**

        $bf' \leftarrow Decode(M')$

        if $bf' < bf_{min}$ **then**

          $V \leftarrow \varnothing$

          $bf_{min} \leftarrow bf'$

        $V \leftarrow V \cup M'$

        $S \leftarrow S \cup M'$

return V

Для оценки вероятности коллизии была проведена симуляция с параметрами: длина информационного слова $\overline{M} = 32$, максимальное число исправляемых БЧХ-кодом ошибок $t = 3$, количество подбираемых ошибок синхронизации $max\, E_{syn} = 3$. В последовательность вносилось до 3 ошибок инверсии и до 5 ошибок синхронизации. В общей сложности было сгенерировано более 15 миллионов последовательностей.

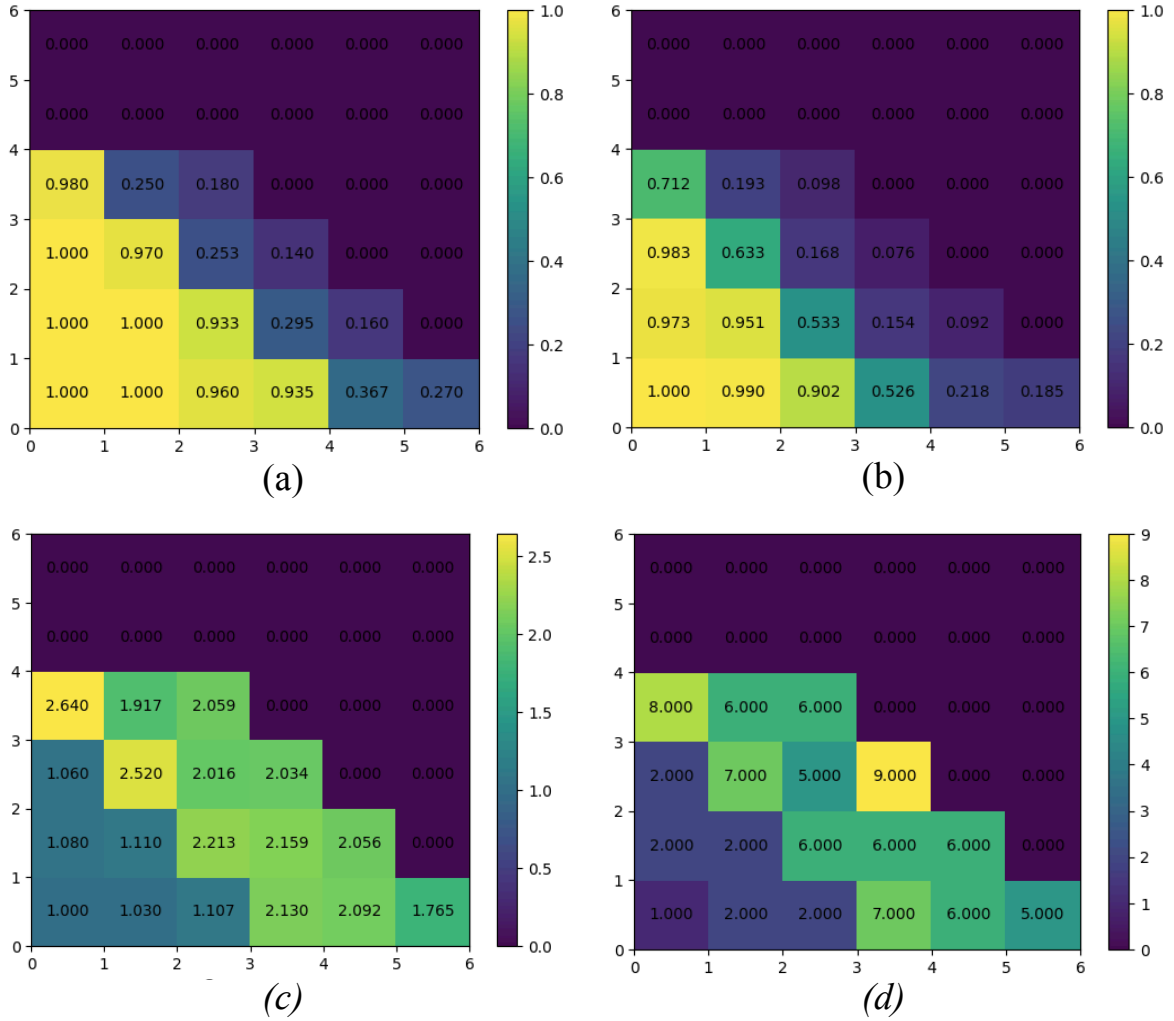


Рисунок 2.7. Тепловая карта: вероятностей присутствия верной последовательности в множестве V (a), вероятностей выбора верной последовательности при выборе первой последовательности из множества V (b), среднего (c) и максимального (d) размера множества V. На вертикальной оси: количество вносимых ошибок инверсии, на горизонтальной оси: количество ошибок синхронизации.

Эксперимент показал применимость БЧХ-кодов для исправления ошибок синхронизации для кодов с указанными параметрами. Код позволяет корректировать суммарно до t ошибок с вероятностью $> 90\%$ (рисунок 2.7a). При суммарном числе ошибок более t отмечается увеличение размера множества V и снижение вероятности нахождения верной последовательности в нем. Итого, алгоритм 2.1 может применяться для исправления ошибок синхронизации вкупе с ошибками инверсии, но не гарантирует нахождение верной последовательности.

2.4 Серверные компоненты системы

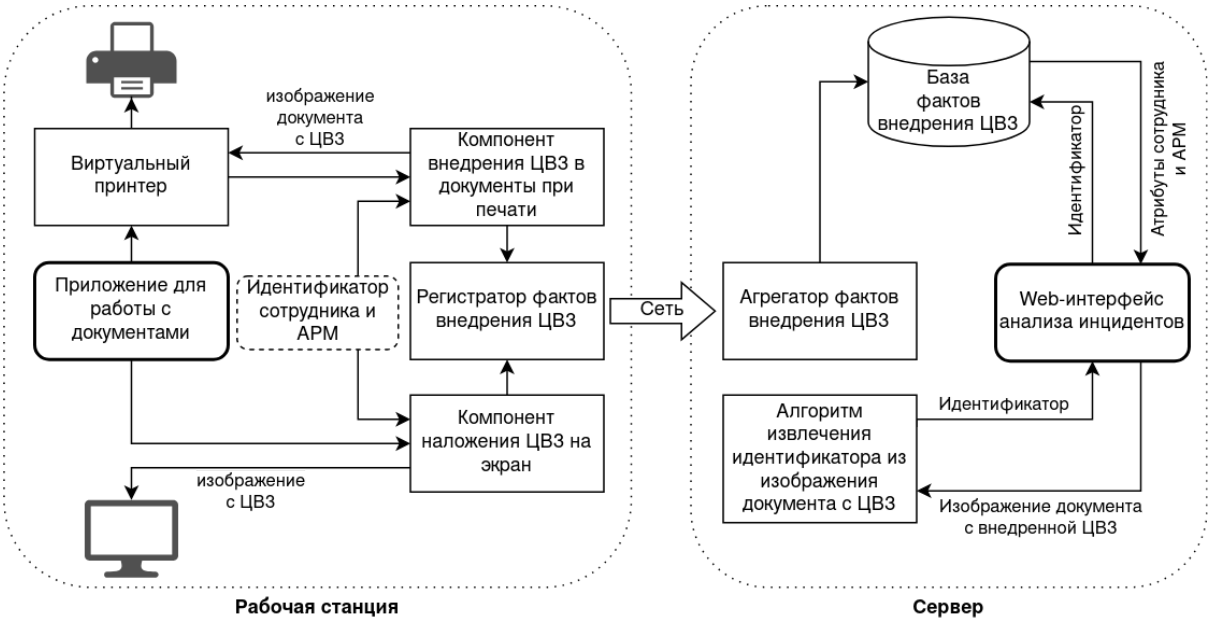


Рисунок 2.8. Схема потоков данных в системе деанонимизации утечек.

Компоненты системы, устанавливаемые на рабочие станции сотрудников, осуществляют внедрение ЦВЗ в текстовые документы при печате и выводе на экран. ЦВЗ содержит идентификатор сотрудника и устройства. Идентификатор состоит из кода департамента и хэш-кода набора атрибутов, определяемого в соответствии с политикой безопасности организации. Идентификатор и соответствующий ему набор атрибутов отправляются в централизованное хранилище: поток данных между клиентскими и серверными компонентами обозначается как *телеметрия*. Телеметрия включает в себя информацию о *фактах внедрения ЦВЗ* в текстовые документы.

Телеметрия передается на сервер посредством сетевых протоколов HTTP/HTTPS к единой точке сбора телеметрии. Для экономии дискового хранилища на сервере и оптимизации поиска при проведении расследования выполняется агрегация фактов внедрения ЦВЗ – объединение множества последовательных событий внедрения в одно путем замены отметок времени на временной интервал.

При расследовании инцидента утечки конфиденциального документа сотрудник службы безопасности загружает подготовленное изображение документа в WEB-интерфейс системы проведения расследований и при помощи алгоритма извлечения получает идентификатор пользователя. По извлеченному идентификатору выполняется поиск в базе данных, и если идентификатор извлечен из документа верно, то в качестве результата аналитику выдается полный набор атрибутов пользователя и рабочей станции.

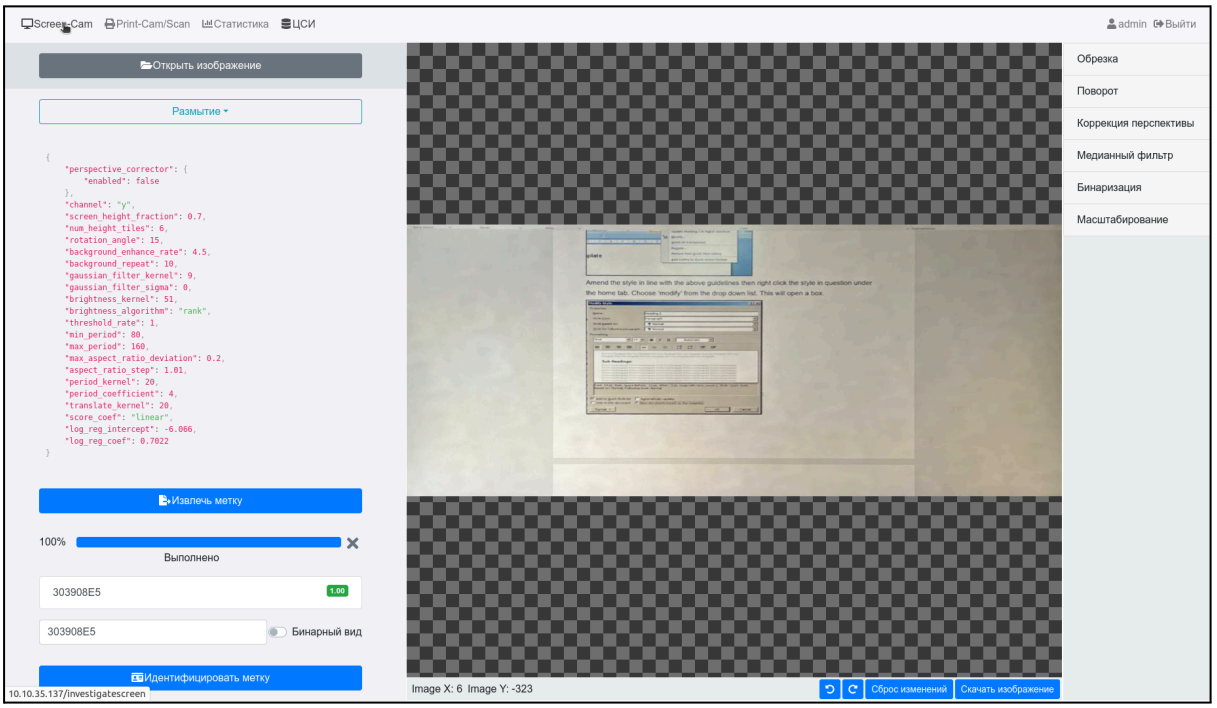


Рисунок 2.9. WEB-интерфейс системы расследования утечек.

Методы внедрения ЦВЗ в текстовые документы предполагают предварительную обработку изображения утечки, поэтому в WEB-интерфейс включен набор инструментов подготовки изображения к извлечению информации из ЦВЗ:

- коррекция перспективы;
- обрезка;
- поворот на произвольный угол;
- медианный фильтр;
- бинаризация (автоматическая или с заданным порогом);

- масштабирование.

2.5 Выводы

Во второй главе описана архитектура системы противодействия анонимным утечкам текстовых документов. Для деанонимизации источника утечки используется внедрение идентификаторов сотрудника и устройства в текстовые документы с помощью ЦВЗ как при печати, так и при выводе на экран. В качестве идентификатора используется бинарная последовательность длины 32, формируемая с помощью хэш-функции из набора атрибутов сотрудника и устройства. Данный идентификатор генерируется на АРМ и передается вместе с атрибутами на сервер в базу данных. В случае утечки служба безопасности проводит расследование – извлекает идентификатор из изображения (утечки) текстового документа с ЦВЗ, и далее, выполняет поиск идентификатора в базе данных и деанонимизирует нарушителя безопасности.

Идентификатор сотрудника и рабочей станции (домен, имя учетной записи, серийный номер диска и другие атрибуты) преобразуется в бинарную последовательность с помощью хэш-функции ввиду ограниченной ёмкости ЦВЗ. Возможны коллизии, при которых два или более сотрудника получают одинаковый идентификатор. В таких случаях при расследовании утечки информации может потребоваться проведение дополнительных действий, таких как проверка журналов логирования или сопоставление данных для точной идентификации нарушителя. Показано, что включение 16-битного идентификатора департамента в 32-битную последовательность снижает вероятность коллизий. С подобной схемой формирования идентификатора сотрудника и устройства вероятность коллизии меньше 5% при количестве сотрудников в организации менее 536 тысяч. Для обнаружения и исправления ошибок при извлечении информации из ЦВЗ используется БЧХ-код.

Алгоритм внедрения ЦВЗ выступает в роли фильтра в конвейере обработки отправленного на печать документа. В ОС семейства Microsoft Windows создается виртуальный XPS-принтер, выполняющий внедрение ЦВЗ в документ и последующее перенаправление документа с ЦВЗ на физический принтер. В ОС семейства Linux модифицируется карта преобразований подсистемы CUPS: в каждый возможный путь добавляется внедрение ЦВЗ. Выбранный механизм внедрения ЦВЗ предполагает работу с документом в растровом формате. Внедрение осуществляется на рабочей станции сотрудника, потенциально обладающей скромными вычислительными ресурсами, непосредственно перед печатью. Для минимизации задержек, обусловленных внедрением ЦВЗ, метод должен обладать низкой вычислительной сложностью.

Механизм наложения ЦВЗ на документы при их выводе на экран реализован с использованием окна-оверлея, отображающего статический водяной знак поверх всех окон, и имеет настраиваемую степень непрозрачности. Данный подход предполагает генерацию ЦВЗ при запуске графической сессии. Отображение статического водяного знака требует минимальных вычислительных ресурсов. Окно-оверлей всегда присутствует на экране, поэтому отображаемый ЦВЗ должен обладать минимально возможной заметностью для повышения комфорта пользователей системы.

Глава 3. Метод внедрения ЦВЗ в текстовые документы при печати

Третья глава посвящена методу внедрения ЦВЗ в текстовые документы при печати. Согласно сформулированной задаче ЦВЗ должны быть устойчивы искажениям, возникающим при печати с последующей оцифровкой фотографированием и отправкой изображения текстового документа через мессенджер, что предполагает сжатие изображения. Для решения задачи выбран класс структурных алгоритмов внедрения ЦВЗ, использующих текстовую разметку документа.

В кратком изложении метод внедрения ЦВЗ в текстовые документы состоит из последовательности шагов:

1. Получение текстовой разметки документа с использованием нейросетевой модели сегментации;
2. Применение фильтров к текстовой разметке, в том числе, для фильтрации рукописного текста и других элементов немашинописного текста;
3. Определение в текстовой разметке информационных блоков;
4. Смещение или перечеркивание слов, кодирующих информацию.

Получение точной текстовой разметки для изображений документов разного качества, в том числе, полученных сканированием или фотографированием, является сложной задачей и представляет собой ключевой этап предлагаемого метода.

Извлечение внедренной в текстовый документ информации осуществляется аналогичным образом, но вместо смещения или перечеркивания выполняется считывание закодированной информации.

Задача получения текстовой разметки по изображению документа сводится к задаче сегментации документа с целью поиска текстовых

элементов на изображении документа. В разделе 3.1 описывается разработанный метод вычисления разметки текстового документа с использованием нейросетевых алгоритмов, являющихся на текущий момент наиболее эффективным подходом в задаче сегментации. Исполнение нейросетевых моделей является ресурсоемкой операцией, поэтому в рамках данной работы также решалась задача оптимизации нейросетевых моделей.

Распечатанные документы могут содержать рукописные надписи и другие элементы немашинописного текста, в том числе печати и подписи. Наличие таких элементов повышает вероятность ошибки при вычислении текстовой разметки изображения, поэтому потребовалась разработка нейросетевой модели детектирования элементов немашинописного текста.

Внедряемый ЦВЗ должен обладать устойчивостью к различным искажениям. Точность извлечения информации из структурного ЦВЗ напрямую зависит от точности вычисления текстовой разметки, поэтому для оценки качества работы алгоритма вычисления текстовой разметки при искажениях была разработана методика тестирования. Методика предполагает имитацию искажений, возникающих при сканировании и фотографировании распечатанных копий текстовых документов, с последующей оценкой устойчивости получаемой разметки.

В разделе 3.2 описаны разработанные структурные механизмы кодирования информации. Первый механизм предполагает горизонтальное смещение слов внутри строки, второй заключается в изменении яркости фрагментов слов машинописного текста. Оба подхода нацелены на достижение минимальной заметности ЦВЗ при сохранении возможности слепого извлечения внедренной в ЦВЗ информации.

Раздел 3.3 содержит выводы по третей главе.

3.1 Разметка текстового документа

Разметка текстового документа – это организованная схема, определяющая расположение и взаимодействие различных текстовых элементов, таких как символы, слова, строки, абзацы, заголовки, списки, таблицы и другие блоки информации. Формально, структура текстового документа может быть представлена как упорядоченное множество:

$$S = \{b_1, b_2, \cdots, b_n\}$$

где задаются текстовые элементы, такие как:

- c_{ij} – символы;
- $w_i = \{c_{i1}, c, \cdots, c_{im}\}$ – слова, состоящие из последовательностей символов;
- $l_i = \{w_{i1}, w_{i2}, \cdots, w_{ik}\}$ – строки, представляющие собой последовательности слов;
- $b_i = \{l_{i1}, l, \cdots, l_{in}\}$ – логические блоки, представляющие собой последовательности строк.

Структура S определяет порядок, взаимосвязь и логическую иерархию этих элементов, обеспечивая восприятие и обработку содержимого документа. Форматирование текста, включая такие параметры, как отступы, выравнивание, размеры шрифтов и стилистические характеристики, может быть выражено функцией форматирования:

$$I = F(S),\ I_{xy} = \{I_{xy1}, I_{xy2}, \cdots, I_{xyC}\},\ I_{xyc} = [0, \cdots, 255],$$

$$0 \le x < W,\ 0 \le y < H,\ 0 \le c < C$$

где функция F задает правила визуального представления для каждого текстового элемента разметки текстового документа S в изображение I (высота H, ширина W, число цветовых каналов C).

Ограничивающий прямоугольник символа можно задать через множество точек, составляющих изображение этого символа. Пусть c — символ, изображение которого представлено множеством точек $P = \{I_{x_1y_1}, I_{x_2y_2}, \cdots, I_{x_py_p}\}$, где каждая точка с координатами (x, y) находится внутри контура символа. Тогда ограничивающий прямоугольник символа c определяется как:

$$R(c) = (x_{min}, y_{min}), (x_{max}, y_{max})$$

где:

- $x_{min} = min_{i=1,\cdots,p} x_i$ – минимальная координата по оси x среди всех точек множества P, которая задает левую границу прямоугольника;
- $y_{min} = min_{i=1,\cdots,p} y_i$ – минимальная координата по оси y среди всех точек множества P, которая задает верхнюю границу прямоугольника;
- $x_{max} = max_{i=1,\cdots,p} x_i$ – максимальная координата по оси x среди всех точек множества P, которая задает правую границу прямоугольника;
- $y_{max} = max_{i=1,\cdots,p} y_i$ – максимальная координата по оси y среди всех точек множества P, которая задает нижнюю границу прямоугольника.

Ограничивающий прямоугольник слова можно определить как минимальный прямоугольник, который полностью охватывает все символы, составляющие это слово. Пусть w обозначает слово, состоящее из символов $c_1c_2, \cdots, c_m$, и каждый символ c_i имеет свой ограничивающий прямоугольник $R(c_i)$. Тогда ограничивающий прямоугольник слова w определяется следующим образом:

$$R(w) = (min_{i=1,\cdots,m} R(c_i)_x, min_{i=1,\cdots,m} R(c_i)_y), (max_{i=1,\cdots,m} R(c_i)_x, max_{i=1,\cdots,m} R(c_i)_y)$$

Аналогичным образом определяются ограничивающие прямоугольники для других структурных элементов разметки текстового документа.

3.1.1 Детектирование текстовых элементов

Задача сегментации изображений является одной из ключевых областей исследований в компьютерном зрении. Сегментация — это процесс разделения изображения на несколько логически связанных сегментов для упрощения последующей обработки. Формально задача сегментации состоит в присвоении каждому пикселю изображения одной или нескольких меток, соответствующих заранее определенным классам. В настоящее время наиболее эффективные методы решения этой задачи основаны на нейронных сетях.

Разметку текстового документа можно рассматривать как задачу сегментации, в которой необходимо определить уровень локализации текстовых элементов. Если целевым уровнем являются символы, то процесс разметки подразумевает их объединение в слова, слова — в строки, а строки — в блоки. Такой подход называется *сегментацией снизу вверх*. Если разметка начинается с крупных элементов (например, блоков текста), которые затем разделяются на строки и слова, такой подход называется *сегментацией сверху вниз*. Возможен также комбинированный подход, который объединяет оба метода.

В данной работе применен комбинированный подход, при котором изначально производится поиск отдельных слов на изображении документа. Этот поиск выполняется с помощью нейронной сети, которая принимает изображение документа на вход и выдает маску с нанесенными ограничивающими прямоугольниками для каждого слова (пример на рисунке 3.1).

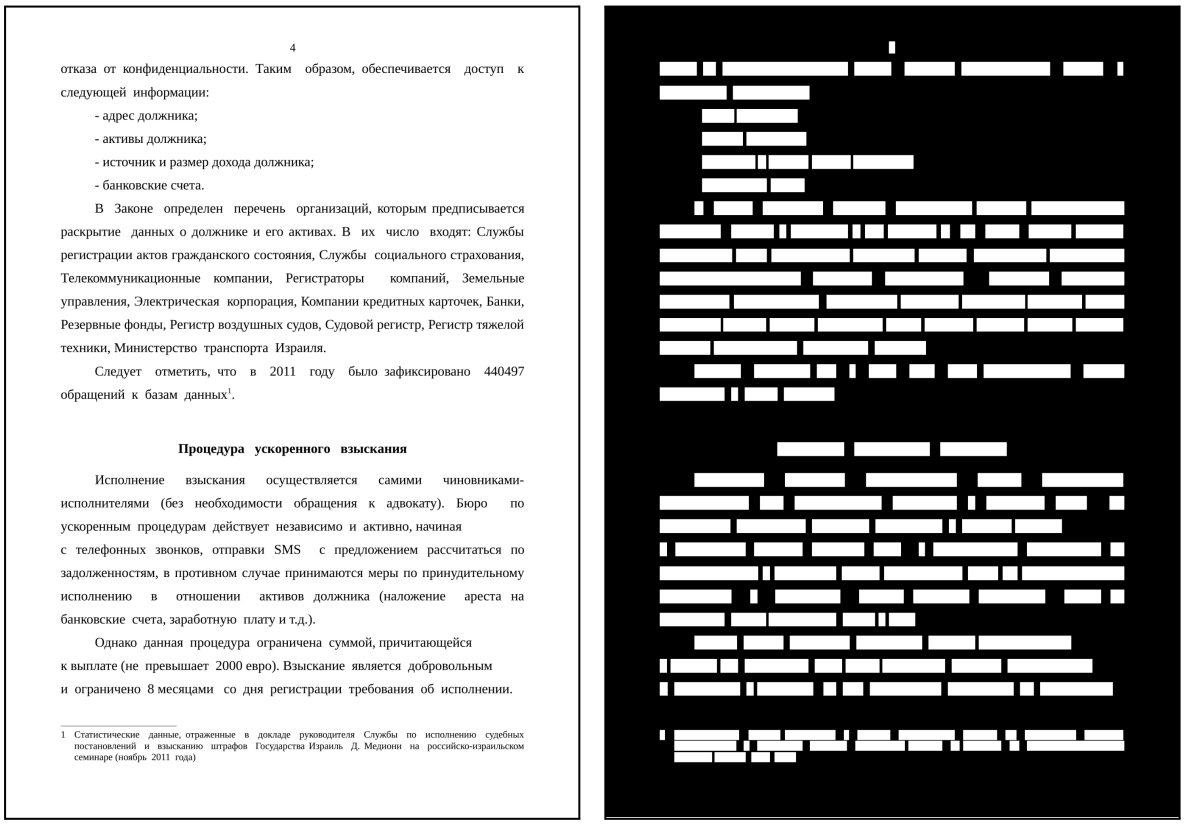
Изображение документа в оттенках серого можно представить следующей функцией:

$$f(x, y) = [0, 255],\ x \in [0, W], y \in [0, H]$$

где W и H это ширина и высота изображения. Требуется найти отображение:

$$G:\ f(x, y) \to g(x, y),\ g(x, y) = \{0, 1\}$$

Каждый пиксель изображения может быть классифицирован как *фон* или *принадлежащий ограничивающему прямоугольнику* (пример на рисунке 3.1).



(a) *(b)*

Рисунок 3.1. Пример входа (a) — изображения текстового документа и выхода (b) — маска ограничивающих прямоугольников слов.

Для эффективного обучения нейросети с учителем требуется большое количество размеченных данных. Часто для обучения используются открытые наборы данных, например, ImageNet [25]. Для рассматриваемой задачи был использован набор данных DDI-100 [66], предназначенный для решения задач обнаружения и распознавания текста в изображениях документов. В его основе более 7000 изображений

уникальных страниц документов, к которым применяются различные искажения и геометрические преобразования: перспективные трансформации, замена фона, сдвиги, наложение текстур, добавление теней, размытие и другие. В результате этих манипуляций на основе одной страницы создается 15 различных версий изображения, что увеличивает суммарный объем набора до более чем 100 000 изображений (примеры на рисунке 3.2). Для каждого изображения имеются аннотации в виде масок текста и местоположений текстовых блоков в формате ограничивающих прямоугольников. Набор данных также включает маски для штампов, которые могут присутствовать на изображениях. Все документы взяты из общедоступных источников и включают отчеты, книги и другие текстовые материалы. DDI-100 разделен на 38 частей, каждая из которых соответствует отдельной книге или отчету. Аннотации содержат координаты текстовых блоков и масок, а также текстовые значения. Изображения документов в DDP-100 визуально близки к целевым изображениям решаемой задачи, предполагающей необходимость обработки изображений документов после сканирования или фотографирования текстовых документов.

Одним из ключевых понятий в машинном обучении является функция потерь, характеризующая разницу между предсказанием модели и истинно верным знанием (Ground Truth). Выбор функции потерь зависит от того, какой результат хочет получить исследователь от обучаемой модели. На практике обычно используют уже хорошо изученные функции потерь либо их модификации.

В рамках данной работы для проведения экспериментов была выбрана составная функция потерь, включающая функцию BCE, функцию IOU и L_1-регуляризацию.

Binary Cross Entropy (BCE) для задачи сегментации — это функция потерь, которая используется при бинарной классификации пикселей изображения для определения класса, к которому принадлежит каждый пиксель.

$$BCE(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}(y_i \, log(\hat{y}_i) + (1 - y_i) \, log(1 - \hat{y}_i))$$

где:

- N — общее количество пикселей в изображении;
- y_i — истинный класс принадлежности пикселя i;
- $\hat{y}_i$ — предсказанная вероятность класса для пикселя i.

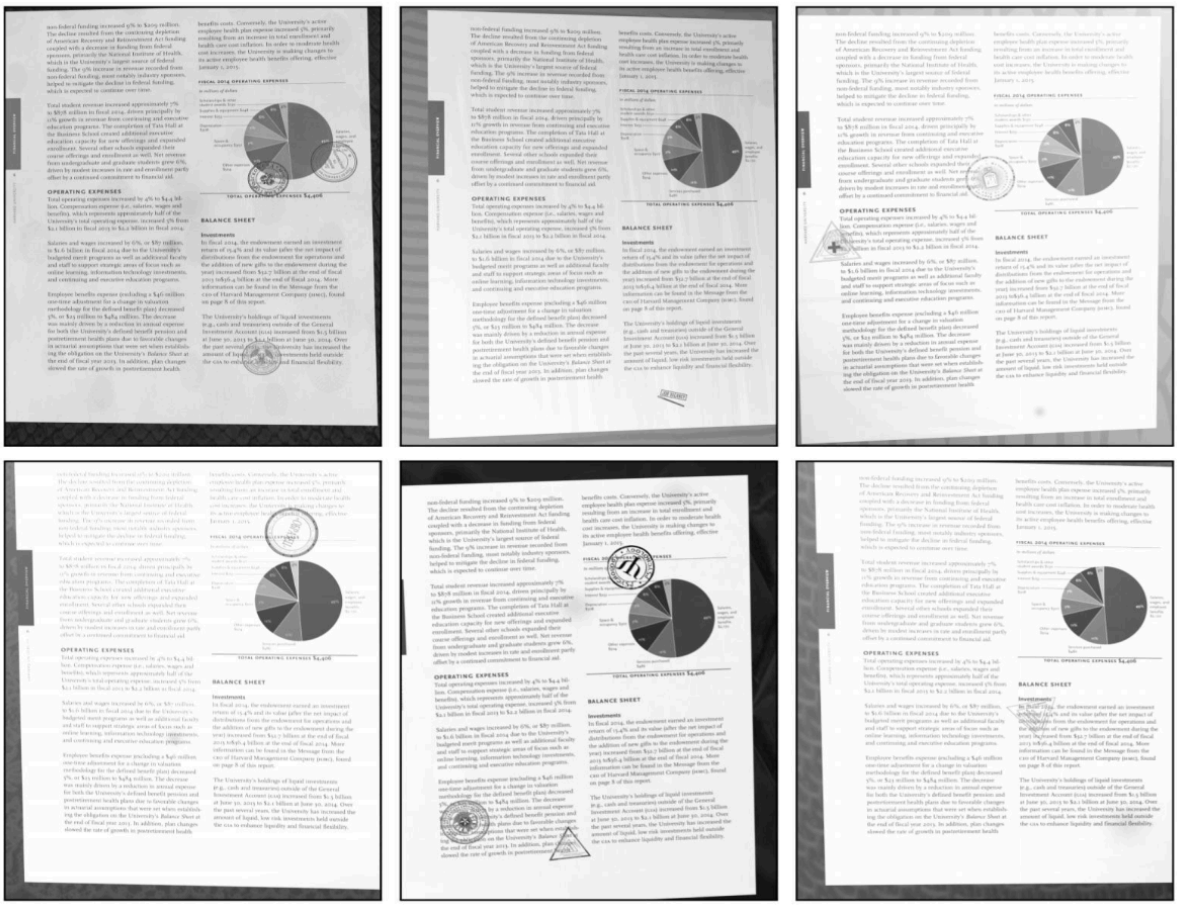


Рисунок 3.2. Примеры изображений текстового документа с различными аугментациями в наборе DDI-100.

Функция потерь на основе *Intersection Over Union* (IOU) в задаче сегментации применяется для оценки сходства между предсказанной маской сегментации и истинной маской. Она измеряет долю пересечения между предсказанными и истинными областями (пикселями) от их объединения:

$$IOU = \frac{Intersection(P,G)}{Union(P,G)} = \frac{|P \cap G|}{|P \cup G|}$$

Здесь:

- P — предсказанная маска (предсказанные классы пикселей);
- G — истинная маска (истинные классы пикселей);
- $|P \cap G|$ — количество пикселей, которые находятся как в предсказанной маске, так и в истинной маске (пересечение);
- $|P \cup G|$ — количество пикселей, которые находятся либо в предсказанной, либо в истинной маске (объединение).

L_1 регуляризация для задачи сегментации — это метод уменьшения сложности модели и предотвращения ее переобучения путем добавления штрафного члена к функции потерь, который основывается на сумме абсолютных значений весов модели. Формально, для модели сегментации с параметрами θ основная функция потерь $Loss(\theta)$, которая измеряет ошибку сегментации при помощи функций BCE и IOU, дополняется регуляризационным членом на основе нормы L_1:

$$Loss_{total}(\theta) = \alpha \cdot BCE + \beta \cdot IOU + \gamma \cdot \sum_i |\theta_i|$$

Для обучения модели, выполняющей разметку изображения текстового документа, использовалась модель на основе архитектуры U-Net [49] — сверточной нейросети, созданной для сегментации биомедицинских изображений. Она состоит из *сжимающей* части, которую называют энкодером (encoder), и *расширяющей* части, которую называют декодером (decoder). Энкодер состоит из повторяющихся

применений двух сверток с ядром размера 3x3, линейного выпрямителя (*ReLU*) для внесения нелинейности и пулинга (*MaxPooling*) для уменьшения пространственной размерности. Декодер же состоит из операций, повышающих пространственную размерность (*Upsampling*), но главной его особенностью является частичное использование промежуточных признаков, выделенных в сжимающей части сети. Данная особенность декодера позволяет получать более точные маски на выходе.

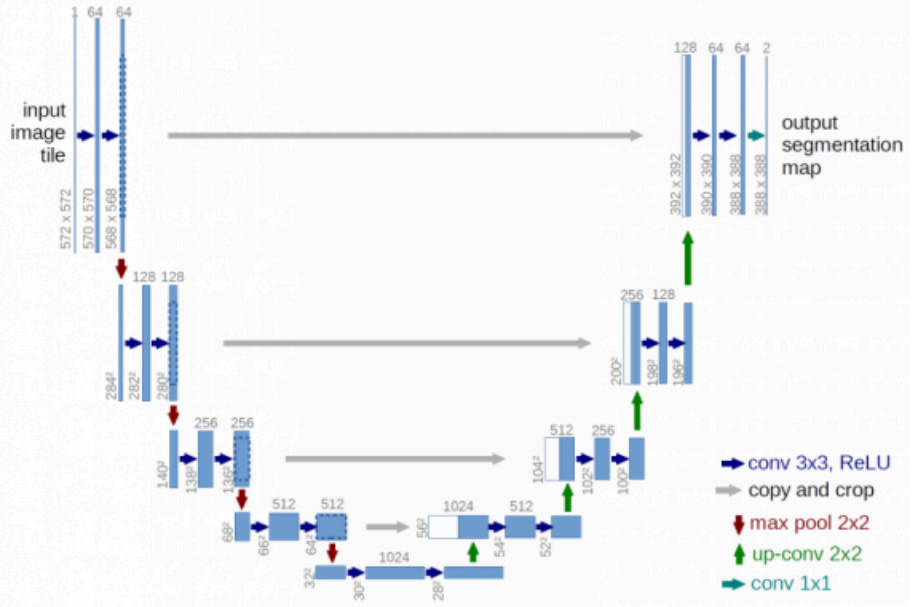


Рисунок 3.3. Схема архитектуры U-Net.

Оценка обученной модели проводилась на специально созданном наборе данных. В качестве исходных данных были использованы PDF-документы с текстовым слоем. Среди документов были представлены технические задания, законы и презентации. PDF-документы конвертировались в изображения, из текстового слоя извлекались ограничивающие прямоугольники символов, выполнялось объединение символов в слова – таким образом были получены оригинальные электронные документы с разметкой слов. Для получения разметки на фотографиях и сканированных изображениях выполнялась следующая последовательность шагов:

1. Распечатка, сканирование и фотографирование с разметкой по страницам;
2. Поиск углов на фотографиях при помощи алгоритма, коррекция перспективы изображения документа по найденным опорным точкам;
3. Наложение разметки исходного текстового документа на изображение документа;
4. Ручное исправление ошибок разметки (пример на рисунке 3.4) если число ошибок сравнительно невелико.

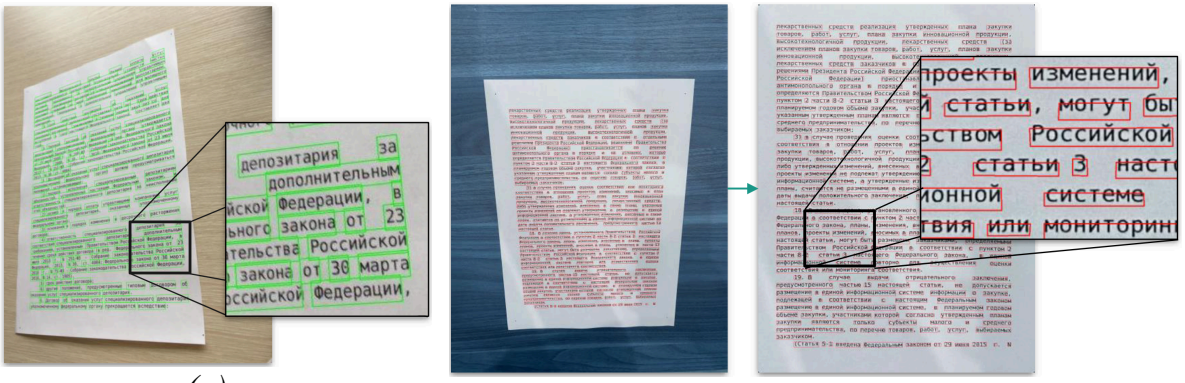


Рисунок 3.4. Проблемы при разметке фотографий и сканов — некорректные ограничивающие прямоугольники слов.

На каждом этапе обработки проводилась ручная валидация результатов автоматической обработки. Изображения документов с большим количеством ошибок отсеивались из набора. Количество документов после отсева на каждом из этапов представлено в таблице 3.1.

Набор данных был разбит на следующие подвыборки в зависимости от условий съемки фотографии:

- *Хорошее качество*: угол наклона от горизонтали <3° и угол наклонения от перпендикуляра к плоскости листа при фотографировании <3° (перспективные искажения);

- *Плохое качество*: угол наклона от горизонтали >3° и угол наклонения от перпендикуляра к плоскости листа при фотографировании >3° (перспективные искажения);
- *Среднее качество*: угол наклона от горизонтали >3° или угол наклонения от перпендикуляра к плоскости листа при фотографировании >3° (перспективные искажения).

Таблица 3.1. Количество изображений документов в наборе данных после ручной фильтрации с разбивкой на этапам обработки.

Номер этапа	*1*	*2*	*3*	*4*
Фотографии	1000	803	803	692
Сканы	398	398	398	339
Всего	1398	1201	1201	1031

Оценка качества текстовой сегментации выполнялась на основе двух метрик: IOU (определена выше) и F_1. F_1-*метрика* для задачи сегментации, где выполняется поиск ограничивающих прямоугольников слов на изображении текстового документа, — это метрика, которая измеряет сбалансированную точность модели, учитывая как точность ($Precision$), так и полноту ($Recall$). В данном контексте, ограничивающие прямоугольники для истинных и предсказанных слов сопоставляются на основе доли пересечения с пороговым значением δ.

Метрика F-score рассчитывается на основе следующих определений:

1. *True Positive (TP)*: Истинное слово было верно сопоставлено с предсказанным словом, если доля пересечения IOU двух прямоугольников превышает порог δ.
2. *False Negative (FN)*: Для истинного слова не нашлось ни одной пары в предсказанных словах, то есть предсказанное слово не пересекается с истинным.

3. *False Positive (FP)*: Слово в предсказании не сопоставляется ни с одним истинным словом, то есть не пересекается с никаким истинным прямоугольником.

Тогда метрика $Precision$ и $Recall$ определяются следующим образом:

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

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

F_1*-метрика* — это гармоническое среднее между $Precision$ и $Recall$:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Эта метрика дает сбалансированное представление о том, насколько хорошо модель находит истинные слова на изображении, одновременно минимизируя как ложные пропуски (FN), так и ложные срабатывания (FP). Значение F-score заключено в отрезке *[0;1]*, где 1 указывает на полное соответствие между предсказанными и истинными прямоугольниками слов.

Для сравнения разработанной нейросетевой модели в задаче текстовой сегментации были выбраны проекты Tesseract OCR [55] (версии 4.1.1) и EasyOCR (версии 1.1.8), представляющие классические и современные подходы к распознаванию текста. Tesseract OCR является популярным инструментом OCR, который использует традиционные методы для выделения текстовых блоков, но может сталкиваться с проблемами при сложных искажения текстов или плохом качестве изображений. EasyOCR использует более современную нейросетевую модель CRAFT (Character Region Awareness for Text Detection) [20] для детекции текста, что позволяет ему эффективно находить текст даже в

сложных условиях, таких как искривления текста и малораспространенные шрифты. EasyOCR демонстрирует высокие результаты на открытых наборах данных. Например, на ICDAR 2015 достигаются точность (Precision) 84.3% и полнота (Recall) 89.8%, а на TotalText [23] — 79.9% точности и 87.6% полноты.

Таблица 3.2. Сравнение реализованного алгоритма текстовой сегментации.

Сценарий	*Модель*	*IOU*	F_1
Фотографии, хорошее качество	Tesseract OCR	0.724653	**0.821935**
	EasyOCR	0.628971	0.641088
	U-Net n4f8	**0.732581**	0.724613
Фотографии, среднее качество	Tesseract OCR	0.346697	0.399617
	EasyOCR	0.598249	0.531117
	U-Net n4f8	**0.651274**	**0.574192**
Фотографии, плохое качество	Tesseract OCR	0.252114	0.323742
	EasyOCR	**0.610773**	**0.591198**
	U-Net n4f8	0.603988	0.510927
Сканированные изображения, черно-белые	Tesseract OCR	0.677850	0.764163
	EasyOCR	0.615979	0.714809
	U-Net n4f8	**0.752524**	**0.817756**

Согласно результатам тестирования на наборе данных, приближенном к условиям эксплуатации программы, модель показывает высокую точность детекции текстовых элементов на изображении документа в сценариях сканирования и фотографирования распечатанной копии документа. В ряде сценариев модель показывает более высокие значения метрик по сравнению с программой EasyOCR, являющейся современным решением на основе глубокого обучения и демонстрирующей высокие результаты на открытых наборах данных. Однако, EasyOCR и большинство современных решений на базе

нейронных сетей ориентированы на работу с использованием видеоускорителей и не предназначены для запуска на процессорах общего назначения, что противоречит требованиям к решению поставленной задачи. Разработанная модель, основанная на основе архитектуры U-Net, состоит из блоков с невысокой вычислительной сложностью и допускает выполнение на процессорах общего назначения.

3.1.2 Оптимизация нейросетевой модели текстовой сегментации

Эксперименты по ускорению алгоритма текстовой сегментации изображения текстового документа дали наилучший результат посредством *дистилляции* и конвертации весов модели. При дистилляции исходная модель использовалась для обучения схожей модели, обладающей меньшим количеством параметров.

В качестве архитектуры для решения задачи текстовой сегментации использовалась архитектура *U-Net*. Одним из ключевых параметров для заданной архитектуры является *количество слоев*. В контексте U-Net слои — это последовательные уровни архитектуры, где происходит обработка входных данных. U-Net состоит из сжимающей и расширяющей частей. Сжимающая часть извлекает иерархические признаки из входных данных, уменьшая пространственные размеры изображения, но увеличивая количество карт признаков. Пусть L — это количество слоев нейросетевой модели, каждый слой $l \in \{1, 2, \cdots, L\}$ включает в себя операции свертки $Conv$ и/или транспонированной свертки $Conv^T$ (или $Upsampling$), и пулинга $Pool$ для уменьшения пространственного размера карты признаков. На каждом уровне l пространство признаков уменьшается на шаге свертки и пулинга в сжимающей части:

$$X^{(l+1)} = Pool(Conv(X^{(l)}))$$

где $X^{(l)}$ — это входная карта признаков на уровне l. В расширяющейся части для восстановления разрешения:

$$X^{(l-1)} = Conv^{T}(X^{(l)})$$

Помимо количества слоев изменялся параметр размера *карты признаков*. Карты признаков — это выходные данные каждого сверточного слоя, представляющие собой результат фильтрации входного изображения. Они содержат информацию об обнаруженных признаках, таких как края, текстуры и более сложные формы на различных уровнях абстракции. Чем глубже в сети слой, тем более сложные признаки представляют карты признаков. В U-Net количество карт признаков обычно удваивается на каждом уровне сжимающей части, а затем уменьшается при восстановлении разрешения в расширяющей части. Пусть $C^{(l)}$ — это количество карт признаков на уровне l. Для слоя l с входным тензором $X^{(l)} \in R^{H^{(l)} \times W^{(l)} \times C^{(l)}}$, где $H^{(l)}$ и $W^{(l)}$ — высота и ширина карты признаков на уровне l, а $C^{(l)}$ — количество карт признаков (глубина тензора). После применения свертки с ядром $K \times K$ и $F^{(l)}$ фильтрами, количество карт признаков становится $C^{(l+1)} = F^{(l)}$, а размер тензора меняется на $X^{(l+1)} \in R^{H^{(l+1)} \times W^{(l+1)} \times C^{(l+1)}}$, где: $H^{(l+1)} = H^{(l)} / 2$, $W^{(l+1)} = W^{(l)} / 2$ — это уменьшение разрешения из-за пулинга. В расширяющей части $C^{(l+1)} = 2 \cdot C^{(l)}$ при каждом новом слое, а в сжимающей части $C^{(l+1)} = C^{(l)} / 2$. Таким образом, если на первом уровне $C^{(1)}$ карт признаков, то через L уровней количество карт на L-м уровне будет $C^{(L)} = 2^{L} \times C^{(1)}$ в сжимающейся части, в расширяющей части это количество будет симметрично уменьшаться.

Дистилляция знаний (knowledge distillation) — это процесс передачи знаний от одной нейросетевой модели (обычно называемой

моделью-учителем) к другой более простой и компактной (называемой *моделью-учеником*), с целью повышения производительности без существенной потери качества предсказаний. Посредством дистилляции осуществлялось обучение моделей-учеников, обладающих иным количеством слоев и начальной размерностью карт признаков.

- $f_T(x)$ — модель-учитель, обученная на наборе данных $D = \{(x_i, y_i)\}_{i=1}^{N}$, где $x_i \in R^d$ – входные данные, а $y_i \in Y$ – истинные метки классов;
- $f_S(x)$ — модель-ученик, которую необходимо обучить.

Основная идея заключается в том, чтобы модель-ученик не только училась на истинных метках данных y_i, как это делается в стандартной постановке задачи классификации, но и воспроизводила выходы (soft labels) модели-учителя, которые могут содержать больше информации о структуре данных. Цель обучения модели-ученика — минимизировать комбинированную функцию потерь:

$$Loss(f_S(x), y) = \alpha \cdot Loss_{hard}(f_S(x), y) + (1 - \alpha) \cdot Loss_{soft}(f_S(x), f_T(x))$$

где:

- $Loss_{hard}(f_S(x), y)$ — стандартная функция потерь на истинных метках y (например, функция перекрестных потерь BCE);
- $Loss_{soft}(f_S(x), f_T(x))$ — функция потерь на «мягких» выходах модели-учителя. Обычно используется кросс-энтропия между распределениями вероятностей (с температурой T, которая сглаживает распределения):

  $$Loss_{soft}(f_S(x), f_T(x)) = -\sum_{k=1}^{K} softmax(\frac{f_T^k(x)}{T})\, log\, softmax(\frac{f_S^k(x)}{T})$$

  где T – температура, используемая для смягчения предсказаний

модели-учителя. При $T > 1$ вероятности распределяются более равномерно, что помогает ученику лучше захватить информацию об относительных вероятностях классов.

- $\alpha \in [0, 1]$ — коэффициент, управляющий балансом между обучением на истинных метках и имитацией поведения учителя.

Также для уменьшения вычислительной сложности исполнения нейросетевой модели использовалась операция преобразования весов модели из 32-битного вещественного типа данных (обозначается далее как $fp32$), в котором производилось обучение, к 16-битному вещественному числу (обозначается далее как $fp16$). Согласно стандарту IEEE754 [29] 32-битное вещественное число состоит из мантиссы размером 23 бита и экспоненты размером 8 бит. При преобразовании к 16-битному вещественного типу размер мантиссы уменьшается до 10 бит, а экспоненты – до 5 бит.

$$fp_{32} = (-1)^{s} \cdot (1 + mantissa_{32}) \cdot 2^{exponent_{32}-127}$$

$$fp_{16} = (-1)^{s} \cdot (1 + mantissa_{16}) \cdot 2^{exponent_{16}-15}$$

В качестве модели-учителя для последующей дистилляции была использована обученная на первом этапе модель архитектуры U-Net с 4 слоями и картой признаков с начальным размером 8. В качестве моделей-учеников также использовались модели архитектуры U-Net. В таблице ниже представлены результаты экспериментов, параметры моделей-учеников закодированы в аббревиатурах вида $nLfC^{(1)}[a\alpha][h]$, где:

- L — количество слоев L модели;
- $C^{(1)}$ — начальный размер карты признаков;
- α — коэффициент, управляющий балансом между обучением на истинных метках и имитацией поведения учителя в функции потерь. Если параметр не указан, то $\alpha = 0$;

- h — флаг преобразования типа весов модели к $fp16$.

Оценка производительности осуществлялась при помощи:

- Моноблок HP TouchSmart 520;
  - Процессор Intel Core i5-2390T – 2.70GHz, 2 ядра, 4 потока, набор инструкций AVX и SSE;
  - Оперативная память DDR3 объемом 4 Гб;
- Операционная система Ubuntu 20.04.5 LTS;
- `GNU time` версии 1.7 — утилита, позволяющая замерять время исполнения команды и ряд других параметров;
- `ONNX Runtime` версии 1.12.1 — кроссплатформенная среда исполнения (inference) моделей машинного обучения (разработчик Microsoft);
- `Python` версии 3.8.10 — высокоуровневый интерпретируемый язык программирования.

Методика оценки производительности включала многократный (100 раз) запуск в отдельном процессе Python-скрипта исполнения модели в связке с утилитой `time`. Измеренные утилитой `time` значения усреднены, на вход подавались тензоры (изображения) размером 1920×1280, заполненные случайными значениями. Исполнение нейросетевой модели в среде *ONNX Runtime* производилось в однопоточном режиме. Время исполнения и потребление памяти включает загрузку и инициализацию интерпретатора `Python`. Также в таблицу включены метрики производительности при запуске скрипта без исполнения нейросетевой модели (строка *No model*).

Метрики производительности:

- *TotalTime* — суммарное время (в секундах), затраченное на исполнение команды;
- *UserTime* — время (в секундах), затраченное процессором на

исполнение команды в пользовательском режиме;

- *Maximum Resident Size (MRS)* — максимальный объем оперативной памяти (в Кб), зарезервированный командой .

С использованием описанного ранее набора инструментов была проведена оценка производительности открытых инструментов текстовой сегментации Tesseract OCR (версии 4.1.1) и EasyOCR (версии 1.7.1) с моделью текстовой сегментации CRAFT.

Таблица 3.3. Сравнение производительности дистиллированных нейросетевых моделей текстовой сегментации.

Модель U-Net	*Производительность*			*Качество*	
	TotalTime	*UserTime*	*MRS*	*IOU*	F_1
n5f8	3.3652	6.287	1266196.48	0.817871	0.970140
n4f8	1.6729	1.781	668537.18	0.869501	0.967100
n4f8h	0.8574	1.077	249896.00	0.838537	0.929569
n4f3	0.7248	0.916	335029.59	0.866140	0.932888
n4f3a0.001	0.7266	0.931	335113.59	0.877489	0.946827
n4f3a0.001h	0.6212	0.854	204846.40	0.877607	0.946645
n4f2a0.1	0.4896	0.705	254094.00	0.859540	0.921917
n4f2a0.01	0.4888	0.701	254140.40	0.862774	0.917592
n4f2	0.4907	0.706	254300.79	0.854980	0.936589
n4f2h	0.3830	0.629	163922.00	0.854267	0.935528
No model	0.2041	0.243	54645.34	-	-

Таблица 3.4. Производительность открытых инструментов текстовой сегментации.

Утилита	*Производительность*			*Качество*	
	TotalTime	*UserTime*	*MRS*	*IOU*	F_1
EasyOCR	123.0433	115.3	3555785.34	0.83468	0.96897
Tesseract OCR	0.9834	0.973	120258.65	0.83053	0.93894

Эксперименты (результаты представлены в таблице 3.3) по оптимизации эталонной нейросетевой модели текстовой сегментации *n4f8*, основанной на архитектуре U-Net, позволили значительно увеличить производительность модели при небольшом снижении качества решения целевой задачи. По результатам экспериментов модели *n4f2h* и *n4f3a0.001h* продемонстрировали наилучшие результаты по соотношению производительности к качеству сегментации. Пиковое потребление оперативной памяти на наиболее производительной модели *n4f2h* снижено в 4.07 раз, время работы снижено в 4.36 раз, качество текстовой сегментации при этом снизилось на 3.26% по метрике F_1 и на 1.75% по метрике IOU. Модель *n4f3a0.001h* показала снижение пикового потребления памяти в 3.26 раз и снижение времени работы в 2.69 раз, качество текстовой сегментации снизилось на 2.1% по метрике F_1 и выросло на 0.91% по метрике IOU.

Открытые инструменты текстовой сегментации Tesseract OCR и EasyOCR продемонстрировали (таблица 3.4) сопоставимые метрики качества F_1 на тестовом наборе данных, однако уступили созданным нейросетевым моделям по времени работы. Кроме того, оценка производительности дистиллированных нейросетевых моделей производилась в однопоточном режиме в среде исполнения *ONNX Runtime*, тогда как Tesseract OCR и EasyOCR не имеют данной настройки и могли использовать все доступные ядра. Инструмент EasyOCR продемонстрировал худшие метрики производительности по времени работы и пиковому потреблению памяти, что объясняется ориентированностью проекта на использование в окружении с графическим ускорителем.

3.1.3 Детектирование рукописного текста

Одной из проблем при текстовой сегментации изображения документа является нестабильная работа на документах с рукописными элементами, такими как инициалы, даты и подписи (пример на рисунке 3.5). Это связано с тем, что инструменты сегментации, используемые для разметки текста, работают непредсказуемо на рукописных фрагментах и других элементах, не относящихся к машинописному тексту. Как показано на рисунке 3.5, машинописный текст выделяется корректно и равномерно, в то время как рукописный текст на обоих изображениях выделен некорректно и по-разному. Для создания компонента разметки, способного корректно обрабатывать документы с рукописными элементами, необходимо внедрить слой фильтрации, который будет исключать из обработки текстовые элементы, отличные от машинописного текста.

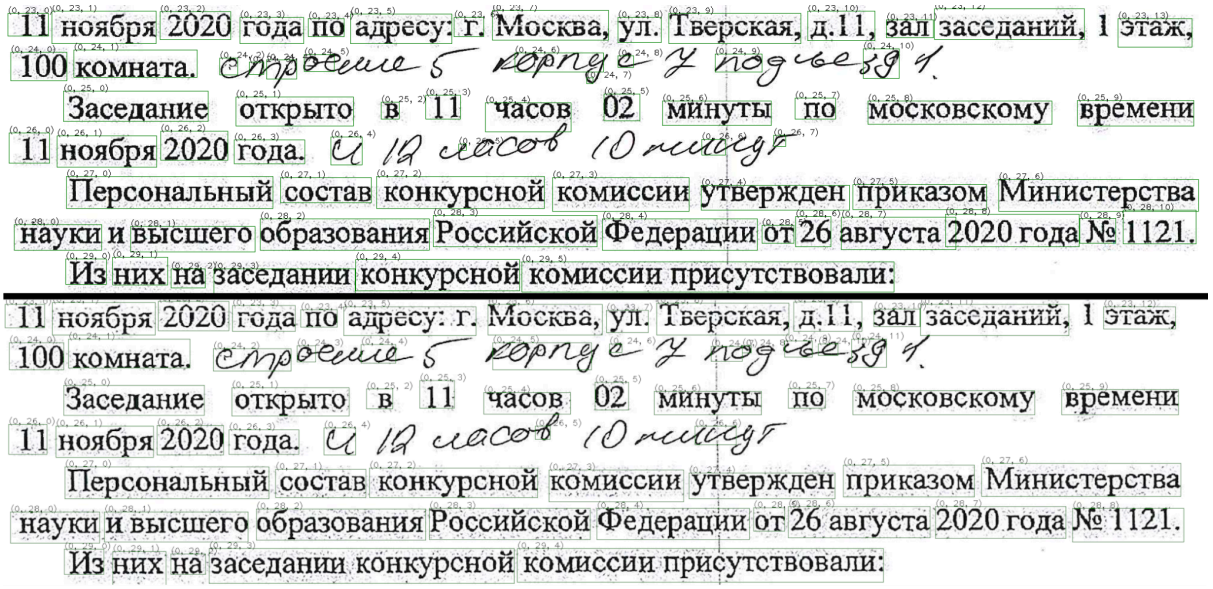


Рисунок 3.5. Пример разметки текста оригинального документа (сверху) и сканированного изображения (снизу).

Для работы с рукописным текстом на изображении документа был разработан компонент детектирования элементов такого типа на основе нейронной сети. Задачу детектирования можно свести к задаче

сегментации, то есть классификации каждого пикселя входного изображения.

Изображение документа в оттенках серого можно представить следующей функцией:

$$f(x, y) = [0, 255],\ x \in [0, W],\ y \in [0, H]$$

где W и H это ширина и высота изображения. Тогда нужно найти отображение:

$$G: f(x, y) \rightarrow g(c, x, y),\ g(c, x, y) = \{0, 1\},\ c \in \{0,\ 1, \cdots, C - 1\}$$

где C — это количество заранее заданных классов, которым может принадлежать каждый пиксель. В текущей задаче $C = 3$, то есть каждый пиксель изображения документа может быть классифицирован как фон, машинописный текст, рукописный текст.

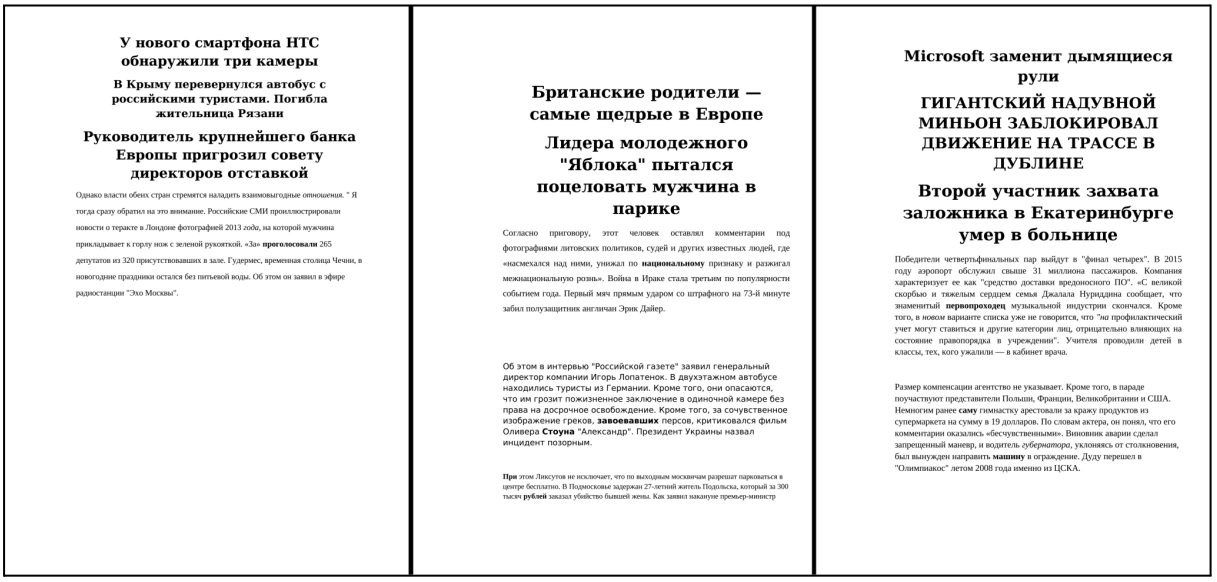


Рисунок 3.6. Примеры сгенерированных документов.

Для эффективного обучения нейросетей необходимо большое количество размеченных данных. В обучении часто используются общедоступные наборы данных, такие как ImageNet [49]. Задача поиска рукописного текста, преимущественно на русском языке, является специфичной и в открытом доступе подходящего набора данных не нашлось. Поэтому был сгенерирован набор данных, подходящий для задачи, на основе DDI-100, состоящий из изображений страниц

электронных документов (статей, книг и т.д.), из которых вырезаны все картинки, графики и другие элементы, и оставлен только машинописный текст. Всего из DDI-100 было использовано 6400 изображений документов.

Также был разработан алгоритм генерации синтетических документов. Входными данными для данного алгоритма являются тексты статей и заголовки, которые были получены с новостных сайтов. Для максимального разнообразия полученных документов, алгоритм случайным образом варьирует следующие параметры:

- Выравнивание текста;
- Величины отступов;
- Наличие заголовков их размер и количество;
- Размер шрифтов;
- Семейство шрифта (наличие засечек, моноширинный);
- Тип шрифта (жирный, курсив);
- Величины межстрочных интервалов;
- Заполненность страницы.

С помощью разработанного алгоритма было сгенерировано еще 6500 изображений документов (пример на рисунке 3.6). Итого получилось 12900 изображений, на которых содержится только машинописный текст.

На реальных документах чаще всего встречается два типа рукописных элементов: *рукописные слова* (фамилии, даты, аббревиатуры и т.д.) и *подписи*. Обычно рукописные слова соразмерны печатному тексту и состоят из отдельных символов. В тоже время размер подписей меньше зависит от размера печатного текста, и редко включают символы. Чтобы в обучающей выборке было представлено оба типа рукописных элементов, было найдено два соответствующих набора данных. Первый [25] состоит из рукописных слов на кириллице. Авторы данного набора данных заявляют, что эти слова были взяты из различных документов, школьных тетрадей и конспектов студентов. Второй набор данных [36] состоит из

подписей, сделанными разными людьми. Набор был собран для обучения моделей способных отличать поддельные подписи от настоящих.

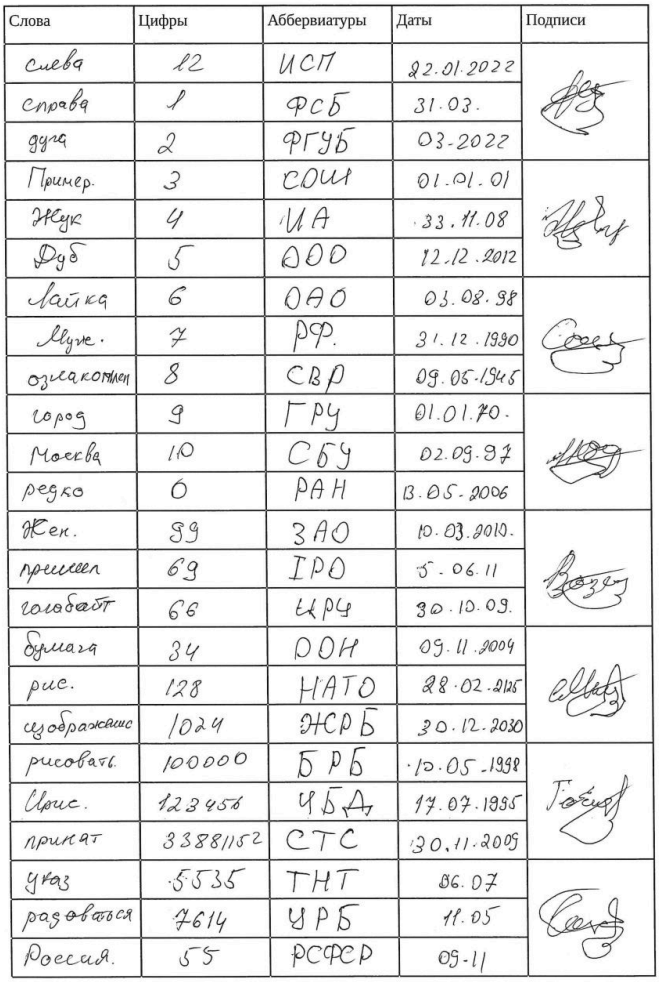
(a) *(b)*

Рисунок 3.7. Примеры из наборов данных рукописных слов (a) и подписей (b).

В дополнение к наборам данных из открытых источников были собраны образцы рукописного текста пяти человек. Для сбора использовался бланк (рисунок 3.8), в котором рукописные элементы были разбиты на пять категорий: слова, цифры, аббревиатуры, даты, подписи.

При генерации набора данных потребовалось решить задачу наложения рукописных элементов на текст документа, визуально схожее с тем, как это сделал бы человек. Были сформулированы следующие правила наложения:

1. Рукописные элементы любого типа не должны полностью пересекаться с машинописным текстом;
2. Рукописные слова должны располагаться вблизи машинописного текста и пересекаться с ним с низкой вероятностью;
3. Рукописные подписи должны располагаться на произвольном удалении от машинописного текста и пересекаться с ним с низкой вероятностью;
4. Рукописные элементы любого типа пересекаются между собой с низкой вероятностью.



Слова	Цифры	Аббервиатуры	Даты	Подписи
слева	12	ИСП	22.01.2022	
справа	1	ФСБ	31.03.	
дуга	2	ФГУБ	03-2022	
Пример.	3	СОШ	01.01.01	
Жук	4	ИА	.33.11.08	
Дуб	5	ООО	12.12.2012	
Майка	6	ОАО	03.08.98	
Муже.	7	РФ.	31.12.1990	
озеракотлет	8	СВР	09.05.1945	
город	9	ГРУ	01.01.70.	
Москва	10	СБУ	02.09.97	
редко	0	РАН	13.05.2006	
Жек.	99	ЗАО	10.03.2010.	
прикеел	69	ИРО	5.06.11	
голобабт	66	ЦРУ	30.10.09.	
бумага	34	ООН	09.11.2004	
рис.	128	НАТО	28.02.2125	
изображаешь	1024	ЖСРБ	30.12.2030	
рисовать	100000	БРБ	10.05.1998	
Цирис.	123456	ЧБД	17.07.1935	
принят	33881152	СТС	30.11.2005	
указ	5535	ТНТ	06.07	
разовается	7614	УРБ	11.05	
Россия.	55	РСФСР	09-11	

Рисунок 3.8. Пример заполненного бланка для сбора рукописных образцов.

Разработанный алгоритм генерации обучающего набора поддерживает три режима наложения рукописных элементов: далеко от машинописного текста, близко к машинописному тексту, с пересечением машинописного текста (степень пересечения конфигурируется). Ниже описаны основные этапы работы алгоритма наложения рукописного элемента на изображение документа.

Алгоритм 3.1. Алгоритм наложения рукописного элемента на изображение документа.

Входные данные: изображение документа I_{doc},

изображение рукописного элемента I_{hw}

Выходные данные: изображение документа с наложенным рукописным элементом I_{doc}^{hw},
маска машинописного текста M_t,
маска рукописного текста M_{hw}

1. Создание маски машинописного текста M_t бинаризацией I_{doc};
2. Создание изображения $\hat{I}_{hw}$ путем масштабирования I_{hw} для соответствия размера элемента с рукописным текстом размеру текста на документе;
   - Высота $\hat{I}_{hw}$ приравнивается средней высоте текстовой линии на I_{doc}
3. Создание грубой маски машинописного текста M_t^{raw} путем применения к изображению I_{doc} операции эрозии с ядром, размер которого равен размеру $\hat{I}_{hw}$;
4. Вычисление маска границ M_{edges}, а также вычисление матрицы градиентов границ M_{edges}^{grad} следующим образом:

$$A = \begin{bmatrix} 0 & -1 & 1 \\ 0 & -2 & 2 \\ 0 & -1 & 1 \end{bmatrix}$$

$$M_1 = M_{edges} \cdot A$$

$$M_2 = M_{edges} \cdot A^T$$

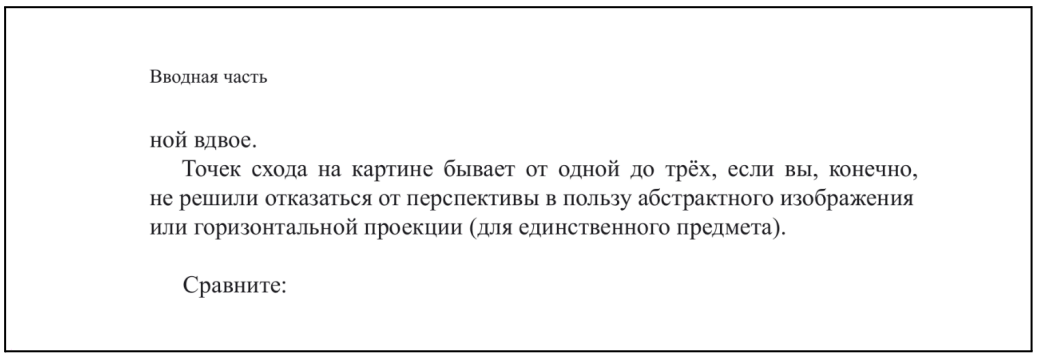
$$M_{edges} = M_1 + M_2$$

$$M_{edges}^{grad} = \left(\frac{M_{1ij}}{\sqrt{M_{1ij}^2 + M_{2ij}^2}}, \frac{M_{2ij}}{\sqrt{M_{1ij}^2 + M_{2ij}^2}} \right)$$

Векторы в матрице M_{edges}^{grad} для каждой точки границы из M_{edges} указывают направление от этой точки к центру текстовой области.

5. Выбор точки наложения $\hat{I}_{hw}$ на I_{doc}:
   - При выборе точки, не принадлежащей маске M_t^{raw}, рукописный элемент будет лежать далеко от печатного текста;
   - При выборе точки, принадлежащей маске M_{edges}, рукописный элемент будет лежать близко к печатному тексту;
   - Для пересечения рукописного элемента с печатным текстом требуется выбрать точку в M_{edges} и переместить ее в направлении соответствующего вектора из M_{edges}^{grad}. Величина перемещения будет определять степень пересечения.
6. Изображение $\hat{I}_{hw}$ накладывается на I_{doc} в выбранной точке, $\hat{I}_{hw}$ накладывается на маску рукописного текста M_{hw}.

(а)


Вводная часть

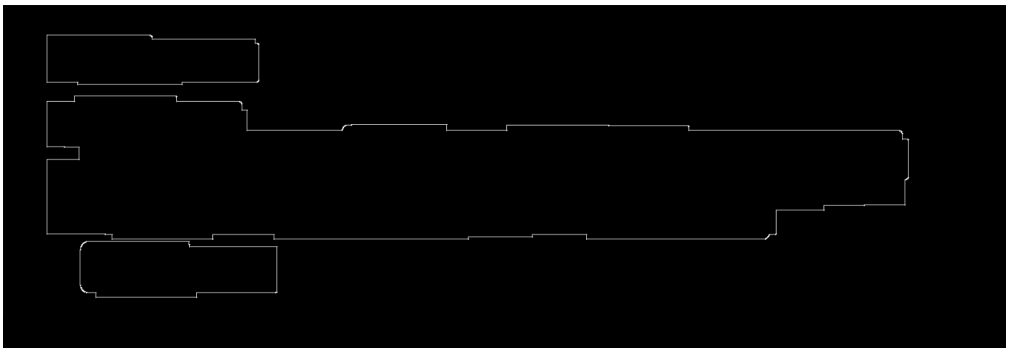
ной вдвое.

Точек схода на картине бывает от одной до трёх, если вы, конечно, не решили отказаться от перспективы в пользу абстрактного изображения или горизонтальной проекции (для единственного предмета).

Сравните:

(b)

(c)



(d)

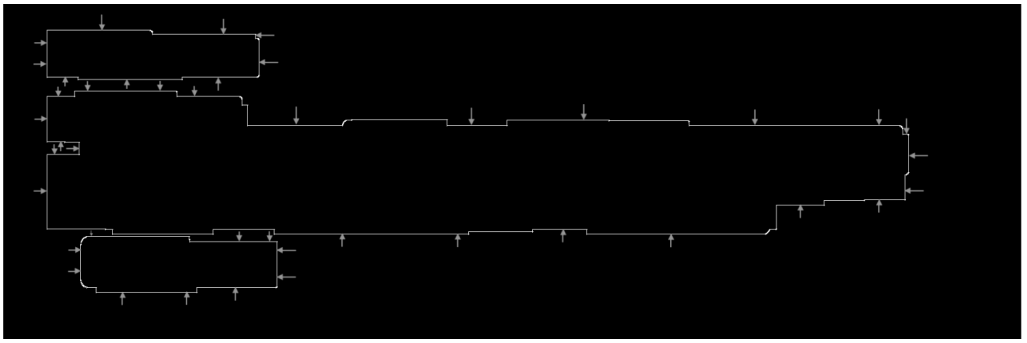


Рисунок 3.9. Этапы предобработки фрагмента документа для наложения рукописных элементов. Оригинальный фрагмент (a), грубая маска машинописного текста M_t^{raw} (b), маска границ (c), маска границ M_{edges}^{grad} с направлениями градиентов (d).

При помощи алгоритма 3.1 на одно изображение документа наносилось по несколько рукописных элементов (количество наложений конфигурируется). Для того, чтобы сбалансировать количество пикселей, принадлежащих машинописному тексту и рукописным элементам, количество накладываемых рукописных элементов было большим, чем встречается на реальных документах. При генерации набора данных для обучения также конфигурировались следующие параметры:

- Количество накладываемых подписей и рукописных слов;
- Величина масштабирования рукописного элемента;
- Вероятностное распределение типов накладывания рукописных элементов: далеко от печатного текста, близко к печатному тексту, с пересечением печатного текста;
- Степень пересечения рукописных элементов с машинописным текстом.

Таблица 3.5. Количественное распределение собранных данных.

Документы из DDI-100	*Сгенерировано*	*Рукописных слов*	*Подписей*	*Размер обучающего набора данных*
6400	6500	899	243	12900

Обучение нейросетевой модели производилось на наборе данных, состоящем из 12900 изображений. Для оценки качества обученных моделей использовалась валидационная выборка из 300 изображений. Оставшиеся 12600 изображений были поделены на обучающую и тестовую выборки в соотношении 80% на 20%. Проверка качества происходит путем вычисления метрики IOU. Данная метрика принимает значения от 0 до 1, большее значение метрики означает лучшее качество модели. Для повышения обобщающей способности результирующей нейросетевой модели при обучении использовались аугментации: фильтр Гаусса, поворот изображения (до 3°), сжатие JPEG.

Эксперименты с обучением модели включали два этапа: выбор функции потерь и выбор архитектуры модели. Выбрана следующая стратегия: сначала определить наилучшую функцию потерь на одной зафиксированной модели, а потом обучать остальные модели с этой функцией потерь. На первом этапе эксперименты с функциями потерь проводились на архитектуре **ures18** (описание в таблице 3.6). Результаты экспериментов представлены в таблице ниже.

Таблица 3.6. Результаты первого этапа экспериментов обучения модели детекции рукописного текста.

Модель	*Функция потерь*	*Количество эпох*	*IOU*
ures18	BCE	10	0.7957
ures18	$DiceLoss$	10	0.9265
ures18	$JaccardLoss$	10	0.9238
ures18	$FocalLoss$	10	0.8175

При обучении были использованы следующие функции потерь. Пусть y — реальный класс объекта, а $\hat{y}$ — предсказание модели, тогда:

$$BCE = -(y\,log(\hat{y}) + (1-y)\,log(1-\hat{y}))$$

$$FocalLoss = -(\alpha(1-\hat{y})^{\gamma} log(\hat{y}) y + (1-\alpha)\hat{y}^{\gamma} log(1-\hat{y})(1-y))$$

$$DiceLoss = 1 - \frac{2y\hat{y}}{y^2+\hat{y}^2}$$

$$JaccardLoss = 1 - \frac{y\hat{y}}{y+\hat{y}-y\hat{y}}$$

Рисунок 3.10. Пример полученных масок для фрагмента документа. Оригинал (слева), DiceLoss (по центру), JaccardLoss (справа).

Наилучшие значения IOU демонстрируют нейросетевые модели, обученные с функциями потерь *DiceLoss* и *JaccardLoss*. Однако, при визуальном анализе масок, сгенерированных моделью (рисунок 3.10), было замечено, что модель, обученная с использованием *DiceLoss*, показывает менее точные результаты в сравнении с моделью, обученной с *JaccardLoss*. В частности, модель обученная с *DiceLoss* чаще допускает ошибки классификации пикселей, расположенных рядом с рукописными элементами. Учитывая этот факт, а также незначительную разницу в

значениях метрики IOU между двумя функциями, в качестве основной функции потерь была выбрана *JaccardLoss*.

Таблица 3.7. Используемые в рамках экспериментов модели.

Обозначение модели	*Описание*
ures18	Модификация модели U-Net, где вместо энкодера используется модель ResNet18
unet5	Модификация модели U-Net с уменьшенным числом промежуточных каналов
unet4	Модификация модели U-Net с уменьшенным числом промежуточных каналов и слоев
uef0	Модификация модели U-Net, где вместо энкодера используется модель EfficientNet b0
umob2	Модель MobileNet v2
umob3	Модель MobileNet v3

На втором этапе экспериментов по обучению нейронной сети для детектирования рукописного текста были использованы модификации известных архитектур, представленные в таблице 3.7. Модели были обучены на небольшом количестве эпох с фиксированной функцией потерь. После первых 10 эпох наилучшие результаты (таблица 3.8) показали модели **unet5**, **unet4** и **ures18**. Обучение продолжилось только для этих моделей до 100 эпох, что оказалось достаточным для их полного обучения. Поскольку разброс значений метрики IOU между моделями был минимальным, окончательный выбор модели для дальнейшего использования основывался на времени обработки. Среднее время запуска на тестовом изображении фиксированного размера указано в таблице 3.9. Эксперименты проводились на процессоре Intel Core i7 6700 с использованием среды выполнения нейронных моделей OpenCV DNN версии 4.6.0. Исходя из результатов экспериментов, модель **unet4** обученная с использованием функции потерь $JaccardLoss$ лучше всего подходит для решения поставленной задачи.

При обучении и валидации нейросети использовались изображения высокого качества, однако на практике документы часто содержат искажения и дефекты печати или сканирования, которые снижают точность детектора рукописного текста. Для решения этой проблемы используется предобработка, которая осветляет изображение для удаления шумов, и постобработка, включающая дилатацию для расширения маски.

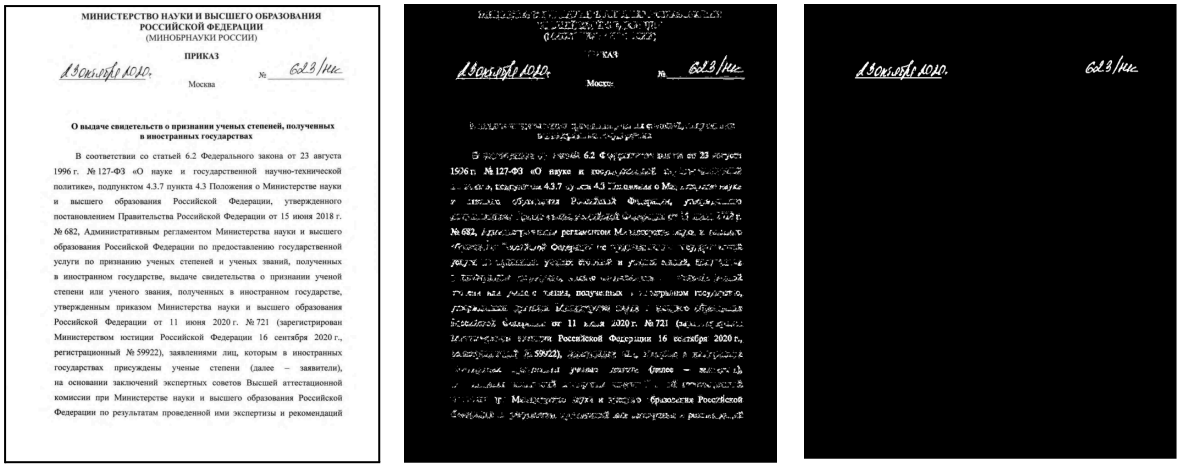
Таблица 3.8. Результаты второго этапа экспериментов обучения модели детекции рукописного текста.

Модель	*Функция потерь*	*Количество эпох*	*IOU*
ures18	$JaccardLoss$	10	0.9238
		100	0.9775
umob3	$JaccardLoss$	10	0.9022
umob2	$JaccardLoss$	10	0.9068
uef0	$JaccardLoss$	10	0.8960
unet5	$JaccardLoss$	10	0.9342
		100	0.9747
unet4	$JaccardLoss$	10	0.9341
		100	0.9739

Таблица 3.9. Среднее время обработки одного изображения.

Модель	*Время обработки, мс*
ures18	1200
unet5	470
unet4	400

Маска рукописного текста, полученная при помощи разработанной нейронной сети, позволяет выполнять проверку текстовых элементов на наличие рукописных элементов, тем самым снижать вероятность ошибок вычисления текстовой разметки документа (пример на рисунке 3.11).



(a) *(b)* *(c)*

Рисунок 3.11. Оригинальный документ (слева), результат работы детектора рукописного текста без предобработки и постобработки (по центру), результат работы детектора с обработкой (справа).

3.1.4 Тестирование метода разметки текстовых документов

Метод нанесения ЦВЗ должен быть устойчив к искажениям, возникающим при печати с последующей оцифровкой документов посредством фотографирования или сканирования. Для эффективного использования структурных методов внедрения текстовая разметка документа после искажений должна иметь минимальное количество различий в сравнении с исходной разметкой. Для оценки *устойчивости* разметки к искажениям была разработана методика оценки данного свойства методов текстовой сегментации. Помимо устойчивости проводилась оценка *точности* разметки – качества текстовой сегментации изображения.

Методика оценки устойчивости метода текстовой сегментации включает следующие этапы:

1. Составление корпуса изображений $I_i \in \{I_1, I_2, \cdots, I_N\}$ документов для оценки;

2. Вычисление текстовой разметки документов $S_i = A(I_i)$ при помощи алгоритма A;
3. Применение искажений $Q(I, \theta)$, не изменяющих геометрию текстовых элементов, к изображениям документов;
4. Применение искажений $P(I, \delta)$, изменяющих геометрию текстовых элементов, к изображениям документов с сохранением параметров искажения δ;
5. Применение искажений $P(S, \delta)$ к разметке документов с параметрами искажения изображений документов δ;
6. Вычисление текстовой разметки $\widehat{S} = A(\widehat{I}_i)$ по искаженным изображениям документов $\widehat{I}_i = P(Q(I_i, \theta_i), \delta_i)$;
7. Вычисление метрик качества $m_i = M(S_i^*, \widehat{S}_i)$ на основе разметки документа до искажения $S_i^* = P(S_i, \delta_i)$ и после применения искажений $\widehat{S}_i$.

Набор искажений предназначен для имитации искажений, возникающих при оцифровке распечатанных документов посредством сканирования или фотографирования. Набор искажений включает:

1. Искажающие геометрию текстовых элементов:
    a. Применение перспективного преобразования;
    b. Применение кусочно-аффинного преобразования;
2. Не искажающие геометрию текстовых элементов:
    a. Наложение эффекта размытия по Гауссу;
    b. Наложение эффекта размытия (motion blur);
    c. Наложение градиента осветления/затемнения;
    d. Наложение бликов;

e. Изменение разрешения изображения документа и применение алгоритма сжатия JPEG.

Перспективное преобразование (код 1a.) применяется для имитации съемки распечатанной копии документа под углом. Перспективное преобразование изображения — это проективное преобразование, моделирующее проекцию трехмерной сцены на двумерную плоскость, где линии, идущие в глубину, сходятся в точке на горизонте, создавая эффект перспективы. Преобразование описывается матрицей размером 3×3, которая действует на однородные координаты точки в двумерном пространстве:

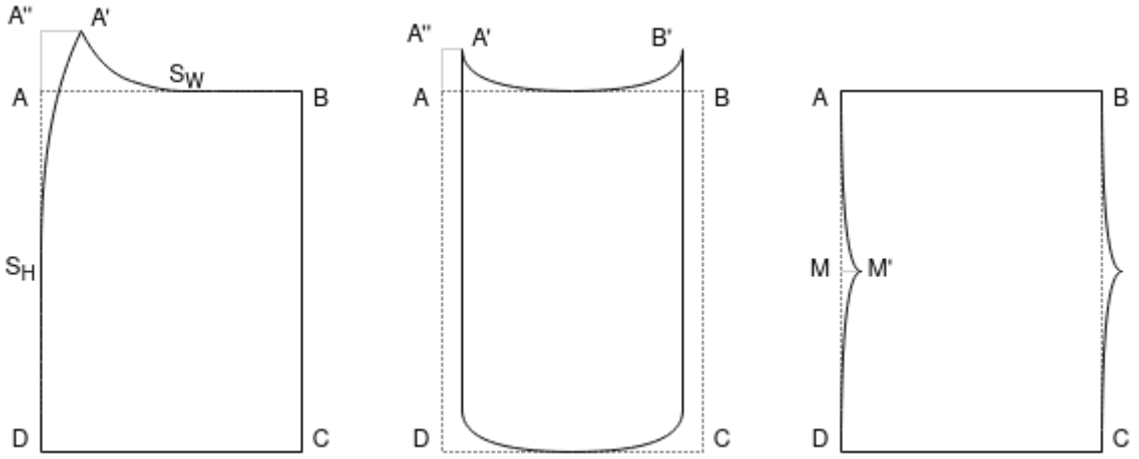
$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Для получения фактических координат, необходимо выполнить нормализацию:

$$x'' = \frac{x'}{w'},\ y'' = \frac{x'}{w'}$$

Перспективное преобразование однозначно определяется четырьмя точками, образующими выпуклый многоугольник. В исходном изображении стороны этого четырехугольника совпадают с границами изображения, а целевая четверка точек смещается случайным образом в диапазонах, заданных коэффициентами, умноженными на высоту r_H или ширину r_W изображения. Матрица преобразования рассчитывается путем решения системы линейных уравнений. Для компенсации изменения масштаба изображение также масштабируется на фиксированный коэффициент s, поскольку после применения перспективного преобразования изменяется масштаб изображения.

Кусочно-аффинное преобразование (код 1b.) позволяет имитировать фотографирование распечатанного документа, расположенного на неровной поверхности, или документа, подвергнутого деформациям.



(a) *(b)* *(c)*

Рисунок 3.12. Схематическое изображение: (a) имитация подогнутого угла листа бумаги, (b) имитация скрученного в трубочку листа бумаги, (c) имитация сложенного пополам листа бумаги.

При имитации подогнутого листа бумаги случайным образом выбираются: параметры изгиба изображения по вертикали AS_H и горизонтали AS_W, а также максимальная высота изгиба AA'' (a) в пикселях. При имитации скрученного в трубочку листа бумаги случайным образом выбираются величина $A'B'/AB$ и высота AA'' (b). Имитация свернутого листа бумаги конфигурируется высотой MM'.

Эффект размытия по Гауссу (код 2a.) – фильтрация изображения, основанная на свертке с ядром Гауссовой функции. При помощи данного эффекта уменьшается детализация изображения. Ядро Гаусса представляет собой двумерную функцию, описывающую нормальное распределение:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

где σ – стандартное отклонение, определяющее степень размытия, задается случайно из диапазона $[0; 2 \cdot \sigma_{max} - 1]$.

Эффект motion blur (код 2b.) — это визуальное размытие объекта или сцены на изображении, вызванное движением объекта, камеры или сцены во время экспозиции. Это приводит к тому, что движущиеся объекты выглядят размытыми вдоль направления их движения. Математически размытие моделируется с помощью свертки изображения с ядром, представляющим траекторию движения объекта. Ядро свертки определяется на основе направления движения, которое задаётся случайно выбранной точкой в диапазоне $[\ -v_{max}, v_{max}]$, где v_{max} – максимальная скорость движения.

Эффект градиента (код 2c.) — это плавное изменение цвета или яркости на изображении от одной точки к другой. Используется для создания эффекта неравномерного освещения документа. В зависимости от случайно выбранного числа создается линейный градиент с максимальным шагом d_{grad}.

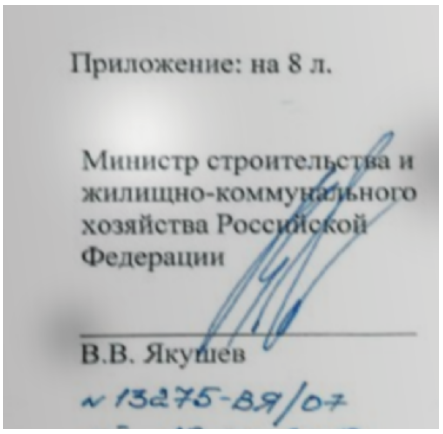


Рисунок 3.13. Пример наложения эффекта блика на изображение документа.

Эффект блика (код 2d.) – это визуальное явление, возникающее, когда яркий источник света отражается на поверхности изображения, создавая яркие, часто разноцветные, пятна. Данный эффект имитируется при помощи изображения градиента, генерируемого при помощи распределения Гаусса, накладываемого на случайную точку изображения документа. Яркость блика – случайное число на отрезке $[\ -v_{max}, v_{max}]$,

его размер – случайное число, определяемое как произведение заданного коэффициента на высоту r_H или ширину r_W изображения. Количество генерируемых бликов k определяется случайным образом.

Изменение разрешения изображения (код 2e.) позволяет имитировать съемку распечатанной копии документа с различного расстояния. Ширина изображения документа выбирается случайно из диапазона $[W_{min}, W_{max}]$, высота изображения изменяется для сохранения исходного соотношения сторон изображения. Изображение кодируется алгоритмом JPEG со случайным образом выбранным минимальным уровнем качества Q_{min}.

В общей сложности было сформировано три профиля искажений (диапазоны значений отражены в таблице 3.10), нацеленных на имитацию искажений в следующих сценариях:

- *Light* – распечатанная копия документа оцифрована посредством сканера высокого качества;
- *Medium* – распечатанная копия документа оцифрована посредством сканера низкого качества;
- *Hard* – распечатанная копия документа оцифрована посредством фотографирования документа и отправлена через мессенджер.

Таблица 3.10. Параметры профилей искажения.

Код искажения	*Параметр*	*Профиль искажения*		
		Light	*Medium*	*Hard*
1a.	*P(1a.)*	0.33	0.66	0.95
	(r_H, r_W)	(0.01, 0.03)	(0.02, 0.05)	(0.03, 0.08)
	s	1.1	1.2	1.4
1b.	*P(1b.)*	0.33	0.66	0.95
	(AS_H, AS_W)	(0.1, 0.3)	(0.1, 0.6)	(0.1, 0.9)
	AA'' (a)	50	100	120

	$A'B'/AB$	(0.2, 0.8)	(0.2, 0.8)	(0.2, 0.8)
	AA'' *(b)*	30	40	55
	MM'	10	20	33
2a.	σ_{max}	2	5	7
2b.	*P(2b.)*	0.33	0.66	0.95
	v_{max}	1	2	3
2c.	d_{grad}	30	60	90
2d.	k	3	10	20
	v_{max}	30	45	60
	(r_H, r_W)	(0.07, 0.3)	(0.07, 0.3)	(0.07, 0.3)
2e.	*P(2e.)*	0.66	0.95	1
	(W_{min}, W_{max})	(1600, W)	(800,1600)	(500,800)
	Q_{min}	60	50	40

Большинство параметров искажений выбираются случайным образом с использованием генератора псевдослучайных чисел. Для обеспечения воспроизводимости результатов инициализация генератора псевдослучайных чисел происходит с фиксированным начальным значением (*seed*), что позволяет при повторных запусках получать одинаковые последовательности случайных значений.

Методика тестирования алгоритмов текстовой разметки документов предполагает применение искажений, изменяющих геометрию текстовых элементов, к исходному изображению документа I_i и разметке текстового документа S_i. Изображение документа подвергается искажениям $P(I_i, \delta_i)$, где δ_i – совокупность параметров после рандомизации для искажений, изменяющих геометрию текстовых элементов. Для получения искаженной

текстовой разметки документа $S_i^* = P(S_i, \delta_i)$ применяется алгоритм искажения разметки, включающий следующие шаги:

1. На основе текстовой разметки документа S_i создается изображение разметки I_{S_i}, состоящее из ограничивающих прямоугольников слов $R(w_j)$, $w_j \in S_i$, заполненных уникальным цветом j, отличным от цвета фона;
2. К изображению разметки I_{S_i} применяются искажения, изменяющие геометрию текстовых элементов;
3. На искаженном изображении разметки $I_{S_i}^* = P(I_{S_i}, \delta_i)$ выполняется поиск каждого искаженного ограничивающего прямоугольника слова $R^*(w_j)$, $w_j \in S_i^*$, имеющего цвет j. Искаженная разметка текстового документа формируется из искаженных прямоугольников слов $S_i^* = \{R^*(w_1), R^*(w_2), \cdots, R^*(w_N)\}$.

Если исходный ограничивающий прямоугольник слова пересекался или был размещен в другом ограничивающем прямоугольнике, то поиск искаженного ограничивающего прямоугольника может быть невозможен. Если количество ошибок при поиске искаженных ограничивающих прямоугольников превышает заданный порог, то формирование искаженной разметки текстового документа считается невозможным.

Таблица 3.11. Результаты тестирования устойчивости разметки текстовых документов.

Модель	*Профиль*	*Документов*	IOU	F_1
U-Net n5f8	Light	208	0.9884	0.9940
EasyOCR	Light	208	0.9837	0.9939
U-Net n4f8	Light	208	0.9869	0.9918

Tesseract OCR	Light	180	0.9903	0.9895
U-Net n4f3a0.001h	Light	208	0.9796	0.9868
U-Net n4f2h	Light	207	0.9765	0.9841
EasyOCR	Medium	208	0.8700	0.9605
U-Net n4f8	Medium	208	0.8834	0.9571
U-Net n5f8	Medium	208	0.8929	0.9553
Tesseract OCR	Medium	179	0.8898	0.9442
U-Net n4f3a0.001h	Medium	206	0.8459	0.9264
EasyOCR	Hard	207	0.8187	0.9253
U-Net n5f8	Hard	207	0.8438	0.9222
U-Net n4f2h	Medium	207	0.8379	0.9115
U-Net n4f8	Hard	204	0.8093	0.9042
U-Net n4f2h	Hard	206	0.7741	0.8428
Tesseract OCR	Hard	178	0.6990	0.7796
Всего				
EasyOCR		623	0.8909	0.9599
U-Net n5f8		623	0.9085	0.9572
U-Net n4f8		622	0.9002	0.9526
U-Net n4f3a0.001h		618	0.8788	0.9394
U-Net n4f2h		620	0.8630	0.9129
Tesseract OCR		537	0.8606	0.9055

Тестирование устойчивости устойчивости алгоритмов текстовой разметки документов осуществлялось на наборе данных, включающем документы из открытых источников. В наборе документов содержатся оригинальные документы и сканированные изображения, включающие рукописный тексты, печати, изображения, графики, таблицы. Общее количество документов 208 изображений. Результаты тестирования представлены в таблице 3.11.

Открытый инструмент разметки EasyOCR, на основе нейросетевой модели CRAFT, показывает одни из самых высоких метрик устойчивости текстовой разметки документов по всем профилям искажения. Однако, данный инструмент ориентирован на работу с использованием графического ускорителя и при использовании процессора общего назначения показывает крайне низкую производительность. Модель *n5f8* демонстрирует близкие результаты в сравнении с инструментом EasyOCR, превосходит в тестах на профиле Light и по метрике IOU, однако, имеет значительно более высокие метрики производительности. Модели на основе U-Net с меньшим числом слоев и карт признаков также продемонстрировали высокие метрики при лучше производительности. Метрики инструмента Tesseract OCR были получены на меньшем числе документов, поскольку для >10% набора документов не удалось создать искаженную разметку текстового документа ввиду большого числа ошибок поиска искаженного ограничивающего прямоугольника (задан порог в 5% от общего числа слов). Инструмент Tesseract OCR показывает устойчивость разметки сравнимую с моделью *U-Net n4f3a0.001h* при искажениях с профилем Light и Medium, но на профиле Hard устойчивость разметки значительно деградировала в сравнении с другими алгоритмами. В итоге, полученные нейросетевые модели демонстрируют устойчивость к искажениям, возникающим при оцифровке распечатанных копии посредством фотографирования и сканирования, сравнимую с лучшими открытыми инструментами текстовой разметки документов.

3.2 Описание структурного метода внедрения ЦВЗ в текстовый документ

Предлагаемый структурный метод внедрения ЦВЗ в текстовые документы использует текстовую разметку документа для кодирования

информации. Использование текстовой разметки обеспечивает устойчивость ЦВЗ к искажениям при печати документа с последующей оцифровкой посредством фотографирования или сканирования. Разработаны подходы кодирования информации, использующие текстовую разметку, при помощи *горизонтального смещения слов* [4] и *перечеркивания слов* [7]. Данный метод позволяют встраивать в текстовый документ ЦВЗ низкой заметности, обеспечивающий деанонимизацию утечек.

3.2.1 Кодирование информации при помощи горизонтального смещения слов

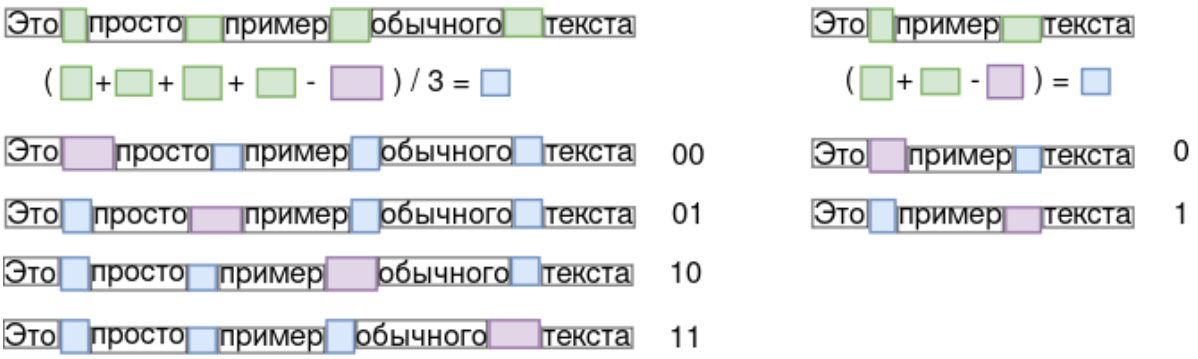
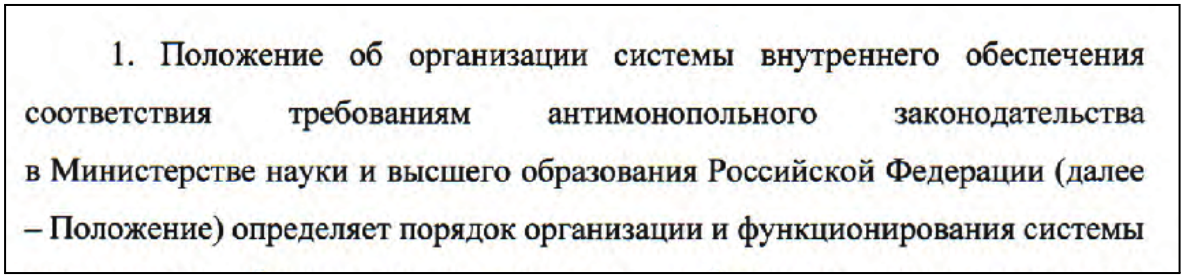


Рисунок 3.14. Схема кодирования информации при помощи горизонтального смещения слов.

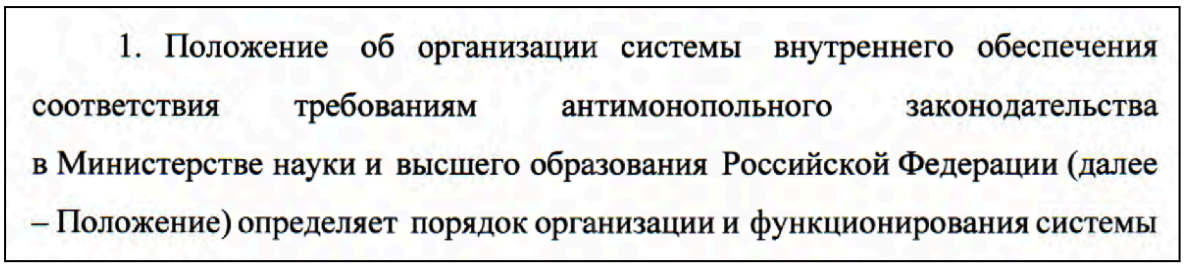
Метод кодирования информации на основе *горизонтального смещения слов* развивает идеи, предложенный в алгоритмах внедрения ЦВЗ в текстовые документы, представленные в работах [41, 19, 22, 38]. Ключевая идея подхода кодирования – горизонтальное смещение слов. Документ разбивается на строки, в каждой из которых посредством жадного алгоритма выделяются блоки последовательно расположенных слов, разделенных четырьмя или двумя пробелами.

При внедрении битовой строки в документ слова горизонтально смещаются таким образом, чтобы величина одного из пробелов в блоке увеличилась, а величина остальных пробелов уменьшилась так, чтобы

общая длина блока осталась неизменной (пример на рисунке 3.14). Позиция увеличенного пробела в блоке позволяет кодировать битовую последовательность: для блоков из четырех пробелов – 2 бита, для блоков из двух пробелов – 1 бит. Стирание ранее внедренной битовой последовтаельности выполняется одновременно с внедрением новой и не требует дополнительных операций. Исходное положение слов в документе восстановлению не подлежит.


1. Положение об организации системы внутреннего обеспечения соответствия требованиям антимонопольного законодательства в Министерстве науки и высшего образования Российской Федерации (далее – Положение) определяет порядок организации и функционирования системы

(a) Оригинальный фрагмент документа


1. Положение об организации системы внутреннего обеспечения соответствия требованиям антимонопольного законодательства в Министерстве науки и высшего образования Российской Федерации (далее – Положение) определяет порядок организации и функционирования системы

(b) Фрагмент документа с внедренным ЦВЗ

Рисунок 3.15. Пример использования механизма кодирования на основе горизонтального смещения.

3.2.2 Кодирование информации при помощи перечеркивания слов

Метод кодирования информации на основе перечеркивания слов имитирует дефекты печати, возникающие при недостаточном уровне чернил в картридже принтера. Метод кодирования изменяет яркость отдельных фрагментов слов. Вдоль горизонтальной оси слова между базовой линий и медианой (рисунок 3.16) выделяется прямоугольная область, и там, где она пересекает символы текста – буквы и некоторые знаки препинания, например, вопросительный/восклицательный знаки –

изменяется яркость. Визуально данный эффект похож на перечеркивание слова осветляющим маркером. Обозначим данное преобразование как нанесение перечеркивающей линии.

Рисунок 3.16. Кодирование на основе перечеркивания слов (слева направо: оригинальное слово, маска, изменённое слово).

1. Положение об организации системы внутреннего обеспечения соответствия требованиям антимонопольного законодательства в Министерстве науки и высшего образования Российской Федерации (далее – Положение) определяет порядок организации и функционирования системы

(c) Оригинальный фрагмент документа

1. Положение об организации системы внутреннего обеспечения соответствия требованиям антимонопольного законодательства в Министерстве науки и высшего образования Российской Федерации (далее – Положение) определяет порядок организации и функционирования системы

(b) Фрагмент документа с внедренным ЦВЗ

Рисунок 3.17. Пример использования механизма кодирования на основе перечеркивания.

При внедрении ЦВЗ документ разбивается на строки. Строки сортируются сверху вниз, слова в строках слева направо. Битовая последовательность внедряется построчно в отсортированный список слов. Для минимизации ошибок (при извлечении в ходе расследования утечки) слова в одной строке объединяются в блоки, кодирующие один бит. Строка делится на фиксированное число блоков. Слова в блоке могут находиться в двух состояниях – перечеркнутом или исходном, – тем самым кодируя 1 бит информации.

Извлечение внедренной бинарной последовательности из изображения документа выполняется построчно. Каждое слово в строке проверяется на наличие перечеркивающей линии, для чего применяется обученная нейронная сеть на основе архитектуры U-Net. Стирание водяного знака выполняется при помощи нейронной сети, восстанавливающей оригинальное визуальное представление каждого слова по отдельности.

3.3 Выводы

В третьей главе описан разработанный структурный метод внедрения ЦВЗ в текстовые документы при печати, удовлетворяющий требованиям:

- устойчивость ЦВЗ к искажениям при печати с оцифровкой посредством сканирования или фотографирования с последующей пересылкой через мессенджеры;
- возможность работы на процессоре общего назначения с минимальным потреблением вычислительных ресурсов;
- возможность слепого извлечения внедренной информации из изображения документа.

В ходе реализации метода была разработана нейросетевая модель сегментации изображения документа для получения его текстовой разметки. Для оценки устойчивости обученной нейросети к искажениям в изображениях документов была разработана методика тестирования. Согласно данной методике разработанная на базе архитектуры U-Net модель *n5f8* проигрывает инструменту EasyOCR на 0.0027 по усредненной метрике F_1 и превосходит на 0.0176 по усредненной метрике *IOU*. Также модель превосходит Tesseract OCR на 0.0517 по метрике F_1 и на 0.0479 по метрике *IOU*. Таким образом, созданная модель построения разметки по

качеству своей работы сравнима со значительно более сложным и ресурсоемким инструментом сегментации EasyOCR и значительно превосходит инструмент Tesseract OCR.

Внедрение ЦВЗ в текстовые документы осуществляется непосредственно перед отправкой документа в физический принтер, поэтому большое значение имеет минимизация задержки, возникающей из-за внедрения ЦВЗ в документ. В результате дистилляции были получены эффективные и точные модели *n5f8* (наиболее точная, но наименее производительная) и *n4f2h* (наименее точная, но наиболее производительная), время исполнения (*UserTime)* разработанных моделей меньше EasyOCR в 18.33 и 183.3 раз соответственно, а пиковое потребление оперативной памяти меньше в 2.8 и 21.7 раз соответственно. Время исполнения (*UserTime)* модели *n4f2h* меньше Tesseract OCR в 1.54 раз, но пиковое потребление оперативной памяти больше на 36.3%.

Разработанные нейросетевые модели используются в структурном методе внедрения ЦВЗ в текстовые документы на основе горизонтального смещения или перечеркивания слов. Фильтр рукописных элементов на основе нейросетевой модели повышает точность разметки изображения. Разработанный структурный метод на основе нейросетевых алгоритмов получения текстовой разметки документа устойчив к искажениям, исполняется на процессоре общего назначения за время, сопоставимое с временем печати одной страницы документа, и поддерживает слепое извлечение внедренной информации.

Глава 4. Метод внедрения ЦВЗ нейросетевым алгоритмом в текстовые документы при выводе на экран

Четвертая глава посвящена методу генерации ЦВЗ нейросетевым алгоритмом для защиты документов при выводе на экран. Разработанный метод предполагает слепое извлечение внедренной информации и обладает свойствами визуальной незаметности и устойчивости к искажениям, возникающим при фотографировании экрана и сжатии алгоритмами, применяемым в мессенджерах.

Выбран подход наложения статического водяного знака поверх документов при помощи окна-оверлея, находящегося поверх всех остальных окон. Водяной знак генерируется при запуске графической сессии пользователя. Такой подход применим к файлам любого формата и ко всем программам для работы с документами, отображающим их содержимое. При использовании окна-оверлея ЦВЗ всегда присутствует на экране.

В данной главе описан принцип работы метода, основанный на использовании *нейронной сети внедрения* для генерации *изображения водяного знака*. Незаметность ЦВЗ обеспечивается свойствами изображения водяного знака, реализуемыми при помощи выбранной функции потерь. Устойчивость реализуется при помощи слоя, имитирующего искажения, возникающие при фотографировании экрана.

Считывание внедренной в ЦВЗ информации выполняется при помощи *нейронной сети определения периода смещения* и *нейронной сети извлечения*. Первая необходима для корректного извлечения *изображения водяного знака* из фотографии экрана, вторая извлекает из усредненного изображения водяного знака внедренную бинарную последовательность. Для извлечения внедренной в ЦВЗ информации оригинальный документ (т.е. документ без ЦВЗ) не требуется.

4.1 Принцип работы предлагаемого метода

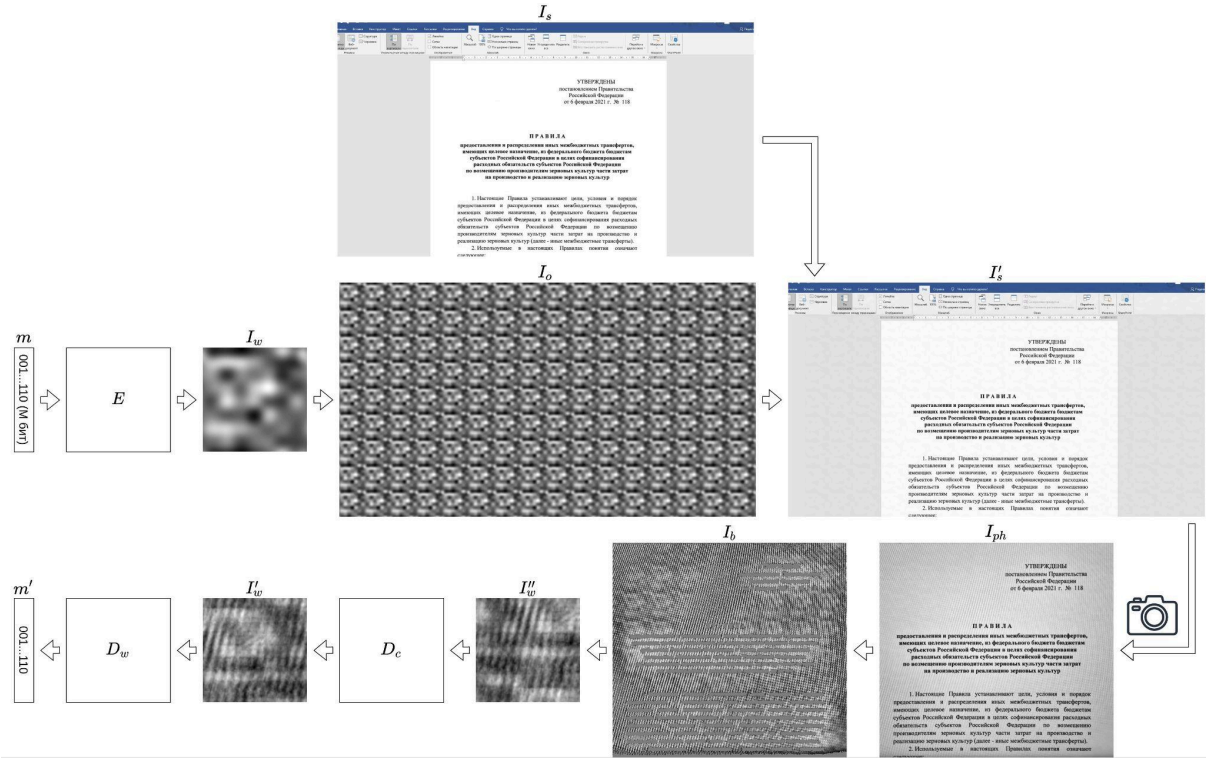


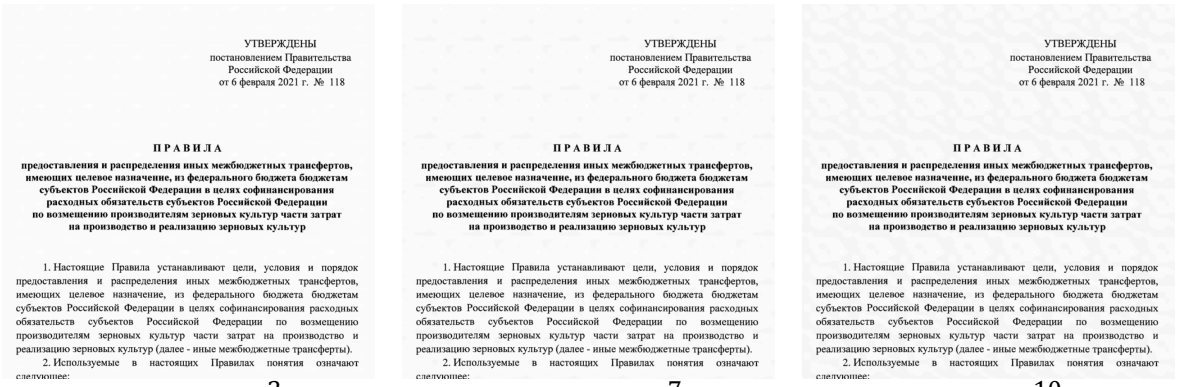
Рисунок 4.1. Схема работы предлагаемого метода.

На рисунке 4.1 схематично представлен разработанный метод внедрения ЦВЗ в текстовые документы при их выводе на экран монитора. На вход *нейронной сети внедрения* E подается *бинарная последовательность* фиксированной длины m, $m_i \in \{0, 1\}$, $i \in \{1, \cdots, M\}$ (метод тестировался при $M = 50$). Нейросеть E формирует *базовое изображение ЦВЗ* I_w в оттенках серого фиксированного размера $S \times S \times 1$ (в реализации $S = 120$) с заданными свойствами:

- высокая степень размытия, благодаря чему отсутствуют резкие переходы цвета между соседними пикселями в I_w;
- отсутствие резких переходов цвета на границах I_w при расположении двух одинаковых изображений I_w либо рядом, либо одного над другим

Исходя из разрешения экрана $W \times H$ формируется *изображение ЦВЗ* I_o, составленное из нескольких изображений I_w, размещенных рядом, в виде сетки. Так, для экрана с разрешением 1920×1080 и $S = 120$ будет составлена сетка, состоящая из 16×9 изображений I_w. Благодаря свойствам изображения I_w, на изображении I_o также будут отсутствовать резкие переходы цвета между соседними пикселями.

Сформированное изображение I_o отображается на экране с некоторой непрозрачностью поверх изображений всех окон графической сессии пользователя в ОС. Результирующее изображение I'_s, выводимое на экран, формируется суммированием изображения I_o с изображением содержимого экрана I_s.



(a) $\alpha = \frac{3}{255}$ *(b)* $\alpha = \frac{7}{255}$ *(c)* $\alpha = \frac{10}{255}$

Рисунок 4.2. Примеры изображений документа с внедренным ЦВЗ с различными значениями коэффициента непрозрачности α.

Важно подчеркнуть, что при создании изображения I_o текущее изображение на экране I_s не используется: I_o остается постоянным вне зависимости от действий пользователя ОС. Это позволяет формировать I_o заранее однократно по бинарной последовательности m и разрешению экрана. Статичность изображения I_o не вызывает дискомфорт при работе с

документами. ЦВЗ всегда присутствует на экране, что позволяет использовать предложенный метод в режиме реального времени.

Двоичная последовательность m' извлекается из фотографии экрана с предварительной коррекцией перспективы и обрезкой областей, не относящихся к экрану. Изображение документа без ЦВЗ не требуется для получения m' (метод предполагает слепое извлечение информации). Благодаря периодичной структуре изображения I_o, водяной знак присутствует во всех областях экрана. При извлечении ЦВЗ переборным алгоритмом определяется величина периода p на фотографии. Изображение I''_w вычисляется как усреднение яркости областей фотографии величиной $p \times p$ с шагом p. Если на фотографии отсутствуют границы экрана монитора, точное положение границ сетки изображений I_w неизвестно, и, следовательно, изображение I''_w может оказаться циклически смещено относительно I_w. Для определения величины этого смещения применяется *нейронная сеть извлечения* D_c. Изображение I'_w получается путем циклического сдвига изображения I''_w на противоположную величину. Затем оно подается на вход *нейронной сети извлечения* D_w, вычисляющей m' .

4.2 Описание архитектуры и процесса обучения нейронных сетей

4.2.1 Нейросеть внедрения E

Нейросеть внедрения E служит для создания на основе бинарной последовательности длиной M бит базового изображения ЦВЗ I_w размером $S \times S \times 1$. E состоит из 2 частей. Сначала M-битная последовательность

подается на вход полносвязного слоя с целью получить тензор, состоящий из S^2 элементов. Полученный тензор приводится к формату $S \times S \times 1$. Его можно интерпретировать как промежуточное представление базового изображения ЦВЗ. Тензор подается на вход второй части нейросети E, выполняющей преобразование полученного тензора к базовому изображению ЦВЗ I_w. Данная нейросеть имеет архитектуру близкую к U-Net [49].

В то же время, важным отличием нейросети E от классической архитектуры U-Net является *циклическое заполнение* (circular padding). Оно определяет характер поведения сверточных слоев на границах сворачиваемого тензора. Граница тензора обрабатывается так, будто в продолжении этой границы находится копия этого тензора. В классической же архитектуре применяется *заполнение нулевыми значениями* (zero padding). Циклическое заполнение используется, например, в задаче обработки фотографий, сделанных в режиме панорамы 360° [61]. Нейронная сеть, принимающая такие изображения, должна учитывать, что области на левой и правой границах фотографии плавно переходят друг в друга, и достигается это применением циклического заполнения по горизонтальной оси. В нейросети E данный прием используется как по горизонтальной оси, так и по вертикальной. Циклическое заполнение позволяет достичь плавного перехода яркости пикселей на изображении I_o.

4.2.2 Нейросеть извлечения D_c

В процессе извлечения бинарной последовательности m' изображение I''_w может оказаться циклически смещено относительно изображения I_w. Для поиска величины циклического сдвига используется нейросеть D_c. Она принимает на вход изображение I''_w, обладающее

размером $S \times S \times 1$. D_c имеет такую же архитектуру, как и вторая часть нейросети E, с применением циклического заполнения. Если подать на вход D_c изображение I_w , будет получен тензор I_c размера $S \times S \times 1$, заполненный следующими значениями:

$$I_c(x,y) = \begin{cases} 1, & (\frac{S}{2} - c \le x \le \frac{S}{2} + c) \cap (\frac{S}{2} - c \le y \le \frac{S}{2} + c) \\ -1, & (0 \le x \le c \cup S - c \le x \le S) \cap (0 \le y \le c \cup S - c \le y \le S) \\ 0, & \text{otherwise} \end{cases}$$

где c — некоторый заданный размер.

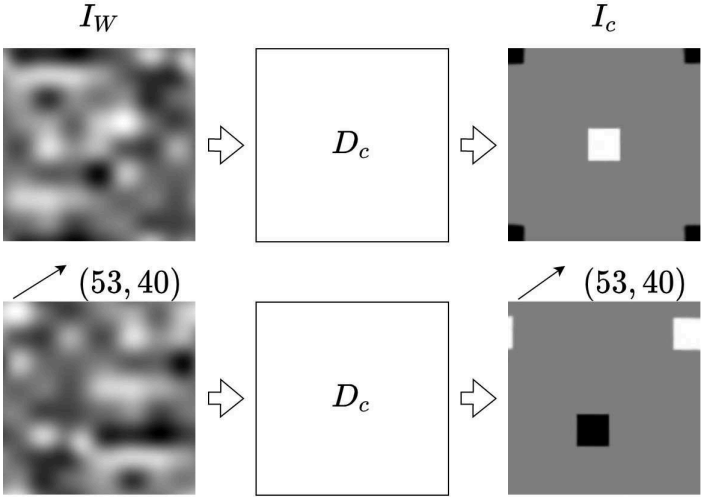


Рисунок 4.3. Инвариантность D_c относительно циклического сдвига входного изображения.

Основное свойство нейронной сети D_c — инвариантность относительно циклического сдвига входного изображения (рисунок 4.3). Так, если применить D_c к изображению I_w, циклически сдвинутому на некоторые значения $(\Delta x, \Delta y)$, на выходе будет получен тензор I_c, сдвинутый на те же значения. Значения сдвига можно определить, осуществив поиск положения максимума выходного тензора относительно его центра. В процессе извлечения ЦВЗ D_c используется для определения циклического сдвига изображения I''_w с целью получить изображение I'_w,

которое отличается от I_w наложением на него некоторого шума, вызванного фотографированием экрана.

4.2.3 Нейросеть извлечения D_w

Задачу извлечения бинарной последовательности m' из изображения I'_w можно интерпретировать как задачу классификации изображений, в которой изображения могут относится одновременно к нескольким классам. Так, если бит m_i равен 1, можно считать это равносильным принадлежности изображения классу под номером i. Этот факт позволяет использовать архитектуры классифицирующих нейронных сетей для нейронной сети извлечения D_w. Для реализации предлагаемого метода была выбрана архитектура *EfficientNet-B2* [58]. Класс архитектур *EfficientNet* был получен применением метода поиска архитектур нейронных сетей NAS (Neural Architecture Search). При одинаковом числе обучаемых параметров нейронные сети *EfficientNet* показывают самую высокую точность по сравнению с другими архитектурами нейронных сетей в задаче классификации на наборе изображений *ImageNet* [25].

4.2.4 Обучение нейронных сетей

Обучение нейронных сетей E, D_c и D_w происходит одновременно. Процесс обучения схематично показан на рисунке 4.4. Каждый шаг оптимизации при обучении состоит из нескольких этапов. Сначала генерируется случайная бинарная последовательность m длины M. По m нейросеть E создает изображение ЦВЗ I_w, над которым выполняется ряд преобразований в *искажающем слое* DL. Полученное изображение I''_w подается на вход нейросети D_c, результатом работы которой является тензор I'_c. По положению максимума тензора I'_c определяется величина

циклического сдвига $(\Delta x, \Delta y)$ изображения I''_w. Изображение I'_w, сформированное обратным сдвигом на $(-\Delta x, -\Delta y)$ изображения I''_w, подается на вход нейросети D_w с целью получить извлекаемую бинарную последовательность m' длины. Затем производится расчет функции потерь $L(m, I_w, I'_c, m')$. При помощи метода обратного распространения ошибки (back propagation) происходит вычисление градиента и обновление параметров нейронных сетей E, D_c и D_w.

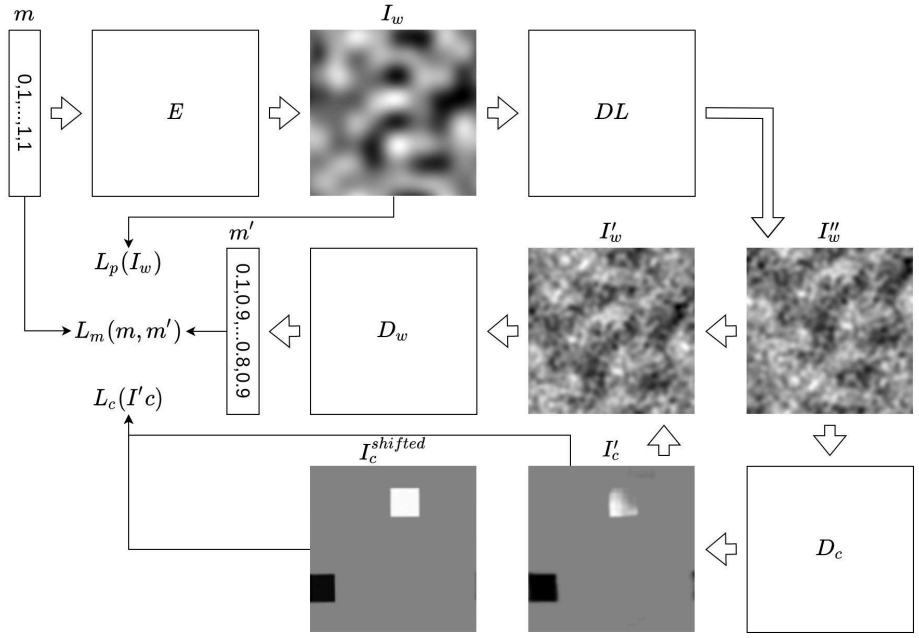


Рисунок 4.4. Схема процесса обучения нейронных сетей.

4.2.5 Искажающий слой *DL*

Искажающий слой DL обеспечивает модификацию изображения I_w. Его основная задача — имитация искажений, возникающих при фотографировании экрана в процессе обучения нейронных сетей. DL состоит из следующих этапов, выполняемых последовательно:

1. Циклический сдвиг на случайные значения $(\Delta x, \Delta y)$;
2. Случайное изменение масштаба изображения в пределах (0.96; 1.04);
3. Поворот на случайный угол в пределах (−2° ; 2°);

4. Добавление Гауссовского шума $\mathcal{N}(0.1)$;
5. Размытие по Гауссу с дисперсией 1

4.2.6 Функция потерь

Функция потерь $L(m, I_w, I'_c, m')$, применяемая в процессе обучения, состоит из трех слагаемых:

$$L(m, I_w, I'_c, m') = \lambda_p L_p(I_w) + \lambda_c L_c(I'_c) + \lambda_m L_m(m, m')$$

где λ_p, λ_c, λ_m — весовые коэффициенты.

Первое слагаемое, функция $L_p(I_w)$, задает свойство плавного перехода яркости на изображении I_w. Каждый пиксель изображения I_w сравнивается с соседними в окне размером 3×3:

$$L_p(I_w) = \sqrt{\frac{1}{S^2}\sum_{x=1}\sum_{y=1}\frac{1}{9}\sum_{\delta x\in\{-1,0,1\}}\sum_{\delta y\in\{-1,0,1\}}(I_w(x,y) - I_w(x+\delta x, y+\delta y))^2}$$

при условиях на x, y:

$$x \notin \{1,\cdots,S\} \Rightarrow x := ((x-1)\ mod\ S) + 1$$
$$y \notin \{1,\cdots,S\} \Rightarrow y := ((x-1)\ mod\ S) + 1$$

Благодаря данным ограничениям плавный переход яркости на изображении I_w обладает свойством цикличности.

Второе слагаемое, функция $L_c(I'_c)$, позволяет обучить нейронную сеть D_c распознавать величину сдвига $(\Delta x, \Delta y)$. Для этого тензор I_c циклически сдвигается на $(\Delta x, \Delta y)$ с учетом обозначенных ранее ограничений на x, y:

$$I_c^{shifted}(x,y) = I_c(x - \Delta x, y - \Delta y)$$

Тензор I'_c сравнивается с полученным тензором $I_c^{shifted}$:

$$L_c(I'_c) = \sqrt{MSE(I'_c, I_c^{shifted})}$$

Третье слагаемое, функция $L_m(m, m')$, отвечает за корректность извлекаемой бинарной последовательности. Как отмечалось ранее, задача извлечения бинарной последовательности из изображения эквивалентна задаче классификации, поэтому для сравнения встроенной последовательности m и извлеченной m' можно использовать функцию бинарной кросс-энтропии:

$$L_m(m, m') = -\frac{1}{M}\sum_{i=1}^{M} m_i log\, m'_i + (1 - m_i)\, log(1 - m'_i).$$

4.3 Алгоритм извлечения информации из ЦВЗ

Фотография документа на экране, попавшая в открытый доступ, используется аналитиком службы безопасности при проведении расследования. От модуля извлечения внедренной в ЦВЗ информации из фотографии экрана (в отличие от модуля встраивания, работающего на АРМ) не требуется высокая скорость работы. Более того, процедура извлечения может быть выполнена неоднократно — с перебором параметров алгоритма. Некоторые этапы извлечения могут быть выполнены аналитиком с применением стороннего ПО обработки цифровых изображений.

Извлечение внедренной в ЦВЗ информации включает шаги:

1. Коррекция перспективы и обрезка частей фотографии, не относящихся к экрану;
2. Выявление фоновых областей в документе на фотографии – далее изображение I_b;
3. Поиск периодичной структуры на изображении I_b, определение периода p;

4. Разбиение изображения I_b с шагом p по горизонтали и вертикали на квадраты размером $p \times p$, усреднение квадратов для получения изображения I_p размером $p \times p$, изменение масштаба изображения I_p для получения изображения I''_w размером $S \times S$;
5. Циклический сдвиг изображения I''_w с помощью нейронной сети D_c, результатом которого является изображение I'_w;
6. Извлечение бинарной последовательности m' из изображения I'_w при помощи нейронной сети D_w

В момент фотографирования плоскость камеры обычно расположена под углом к плоскости экрана, что приводит к искажению перспективы изображения на экране. На первом этапе извлечения необходимо выполнить коррекцию перспективы фотографии: ее выполняет аналитик с применением графического редактора. Для этого на фотографии определяются направляющие линии, которые на скорректированном изображении должны быть расположены горизонтально или вертикально. Так, в качестве направляющих линий могут использоваться границы экрана монитора на фотографии или, если границы оказались за пределами фотографии, элементы изображения, в том числе границы окон программ и/или строки текста сфотографированного документа. Также в графическом редакторе выполняется обрезка скорректированной фотографии с целью оставить на итоговом изображении только области, относящиеся к изображению на экране.

Дальнейшая обработка фотографии происходит в канале яркости Y, полученном по каналам красного, зеленого и синего цветов согласно стандарту [46]. Обозначим монохромное изображение в этом канале как I_{ph}

. Для определения фоновых областей в документе на изображении I_{ph} (второй шаг алгоритма извлечения) значение яркости в каждом пикселе сравнивается с медианным значением в окне заданного размера $a \times a$ с центром в этом пикселе. Изображение I_b рассчитывается следующим образом:

$$I_b(x,y) = \begin{cases} I_{ph}(x,y) - I_{ph}^a(x,y), & |I_{ph}(x,y) - I_{ph}^a(x,y)| \le t \\ 0, & |I_{ph}(x,y) - I_{ph}^a(x,y)| > t \end{cases}$$

где $I_{ph}^a(x,\ y)$ – медианное значение I_{ph} в окне размером $a \times a$ с центром в $(x,\ y)$, t – заданное пороговое значение.

Как правило, изображения текстовых документов представляют собой набор символов на однотонном, преимущественно белом фоне. Благодаря этому свойству в областях фона изображение I_b близко к фрагменту встроенного изображения I_o с измененным масштабом. Цель третьего этапа извлечения — выявить периодичную структуру на изображении I_b и определить величину периода p. Обозначим за I_p усреднение изображения I_b с шагом p:

$$I_p(x,y) = \frac{1}{\left\lfloor \frac{W}{p} \right\rfloor \cdot \left\lfloor \frac{H}{p} \right\rfloor} \sum_{k,l=0}^{\left\lfloor \frac{W}{p} \right\rfloor - 1, \left\lfloor \frac{H}{p} \right\rfloor - 1} I_b(k \cdot p + x, l \cdot p + y)$$

Определение периода основано на следующем наблюдении. Если величина p не соответствует искомому периоду, распределение значений пикселей изображения I_p оказывается близко к случайному шуму. После устранения шума с помощью фильтра Гаусса $G(\cdot)$ изображение мало отличается от однотонного. Среднеквадратичное отклонение $std(\cdot)$ при этом близко к 0. Если же p совпадает с периодом, применение фильтра

Гаусса к I_p сохраняет переходы яркости и среднеквадратичное отклонение оказывается значительно выше нуля. Таким образом, поиск периода можно осуществить перебором по диапазону (конфигурируемый параметр алгоритма извлечения) значений $p \in [p0, \cdots, p1]$:

$$p = arg\,max_{p' \in [p_0, \cdots, p_1]}\, std(G(I_{p'}))$$

Масштаб изображения I_p, полученного при усреднении I_b, меняется на $S \times S$, в результате чего формируется изображение I''_w. Дальнейшие шаги (5 и 6) извлечения бинарной последовательности m' из изображения I''_w совпадают с обработкой изображения I''_w при обучении нейронных сетей (раздел 4.2).

4.4 Выводы

В четвертой главе описан разработанный метод внедрения ЦВЗ в текстовые документы, выводимые на экран, удовлетворяющий требованиям:

- возможность слепого извлечения внедренной в ЦВЗ информации;
- ЦВЗ на экране не должен вызывать дискомфорта у пользователей;
- устойчивость к искажениям, возникающим при фотографировании выведенного на экран документа с последующей отправкой через мессенджер.

Получение внедренной в ЦВЗ информации осуществляется при помощи извлекающей нейросети D_w, принимающей на вход усредненное изображение ЦВЗ с коррекцией смещения и не использующей изображение оригинального документа (без ЦВЗ). Изображение ЦВЗ генерируется нейросетевой моделью E, функция потерь при обучении включала компонент, обеспечивающий высокую степень размытия и плавные переходы между пикселями. Устойчивость ЦВЗ к искажениям

обеспечена применением искажающего слоя *DL* при обучении. Подробное описание результатов тестирования метода приведено в разделе 5.2. Разработанный статический метод генерации ЦВЗ на основе нейросетевых алгоритмов обеспечивает комфортную работу с документами для пользователей и устойчив к искажениям при фотографировании экрана, что положительно отличает метод от существующих.

Глава 5. Тестирование системы противодействия анонимности утечек текстовых документов

Пятая глава посвящена тестированию реализованной системы противодействия анонимности при утечках текстовых документов. Система должна обеспечивать внедрение в текстовые документы информации, позволяющей устанавливать виновников публичных утечек. Объектами тестирования являются разработанные методы внедрения ЦВЗ в документы при печати и выводе на экран.

Раздел 5.1 содержит описание и результаты тестирования метода внедрения ЦВЗ в текстовые документы при печати, предполагающего слепое извлечение встроенной информации. Методика тестирования позволяет оценить устойчивость метода к различным искажениям и преобразованиям, сопутствующим печати документа с последующей оцифровкой посредством сканирования или фотографирования.

Раздел 5.2 содержит описание и результаты тестирования метода внедрения ЦВЗ в текстовые документы при выводе на экран, предполагающего слепое извлечение встроенной информации. Методика тестирования позволяет оценить устойчивость метода к различным искажениям и преобразованиям, сопутствующим фотографированию выведенного на экран документа с последующей отправкой фотографии документа через мессенджер.

Раздел 5.3 содержит выводы по пятой главе.

5.1 Тестирование метода внедрения ЦВЗ при печати

Структурный метод внедрения ЦВЗ в текстовые документы на основе горизонтального смещения или перечеркивания слов был реализован и протестирован. Внедряемая бинарная последовательность длины 50 включает 32-битный идентификатор сотрудника и устройства и

18 бит БЧХ-кода для обнаружения и исправления 3 или менее ошибок. Будем считать, что бинарная последовательность извлечена корректно, если в извлеченной 50-битной последовательности не более 3 ошибок.

Экспериментальная оценка разработанного метода [3] осуществлялась на основе качественной и количественной оценок основных параметров ЦВЗ:

- *емкость* – количество информации, которое может быть внедрено в текстовый документ посредством разработанного метода встраивания ЦВЗ;
- *незаметность* – оценка заметности искажений изображения текстового документа в результате внедрения ЦВЗ и стойкости метода внедрения ЦВЗ к стеганографическому анализу;
- *устойчивость* – оценка разработанного метода извлечения ЦВЗ к искажениям, возникающим при фотографировании или сканировании распечатанного текстового документа с внедренным ЦВЗ.

5.1.1 Оценка емкости текстовых документов при использовании структурного метода внедрения ЦВЗ

Емкость текстового документа – величина, характеризующая максимально возможное количество информации, которое может быть встроено в текстовый документ при помощи структурного метода внедрения ЦВЗ. Емкость документа η_{shift} при внедрении ЦВЗ с использованием метода на основе горизонтального смещения слов рассчитывается по формуле:

$$\eta_{shift} = \sum_{i=1}^{N} \lfloor \frac{|l_i|-1}{2} \rfloor,\ l_i \in S,$$

где N – количество строк в документе.

При использовании структурного метода внедрения ЦВЗ на основе перечеркивания слов, емкость документа η_{strike} рассчитывается по формуле:

$$\eta_{strike} = \sum_{i=1}^{N} |l_i|.$$

Исходя из требований к электронным документам, составленным согласно ГОСТ Р 7.0.97–2016 [6], электронный документ формата А4 может содержать от 52 (межстрочный интервал 1, кегль шрифта 12 пт) до 30 строк (межстрочный интервал 1.5, кегль шрифта 14 пт), документ формата А5 – от 34 (межстрочный интервал 1, кегль шрифта 12 пт) до 19 строк (межстрочный интервал 1.5, кегль шрифта 14 пт). Среднее число слов в строке оценивается от 10 (кегль шрифта 12 пт) до 8 (кегль шрифта 14 пт) для документа формата А4, для документа формата А5 – от 7 (кегль шрифта 12 пт) до 5 (кегль шрифта 14 пт).

Таким образом, электронный документ формата А4 имеет емкость от 90 до 208 бит при внедрении ЦВЗ структурным методом на основе горизонтального смещения слов. При использовании метода на основе перечеркивания слов емкость документа оценивается от 240 до 520 бит. Для документов формата А5 емкость документа от 38 до 102 бит при использовании метода на основе горизонтального смещения слов, для метода на основе перечеркивания слов от 95 до 238 бит.

Полученные значения емкости документа возможны при полном заполнении текстом страницы электронного документа. На практике маловероятно, что все страницы документа будут полностью заполнены текстом, поэтому фактическая емкость документов ниже. При недостаточной для внедрения ЦВЗ емкости документа на печать отправляется исходный документ.

5.1.2 Оценка незаметности и стойкости к стеганографическому анализу структурного метода внедрения ЦВЗ

Для оценки *незаметности* разработанного метода внедрения ЦВЗ на основе горизонтального смещения и перечеркивания слов применялись метрики PSNR и SSIM. Данные метрики изначально использовались для оценки качества алгоритмов сжатия с потерями. Метрика PSNR (Peak Signal to Noise Ratio) определяется как пиковое отношение сигнала к шуму и измеряется в децибелах. При расчете значения данной метрики используется среднеквадратичное отклонение MSE (Mean Squared Error). Метрики вычисляются по формулам:

$$MSE(I, I_w) = \frac{1}{W \cdot H} \sum_{x=1}^{W} \sum_{y=1}^{H} (I(x,y) - I_w(x,y))^2$$

$$PSNR(I, I_w) = 10\ log_{10} \frac{MAX_I^2}{MSE(I, I_w)}, MAX_I = 255$$

Чем выше значение PSNR, тем ближе изображение с ЦВЗ к исходному. Типичные значения PSNR для алгоритмов сжатия изображений лежат в пределах от 30 дБ до 50 дБ [51], причем значения от 40 дБ считаются высокими. Существенный недостаток PSNR — при расчете не учитывается пространственное расположение пикселей изображения. Для решения этой проблемы используется индекс структурного сходства SSIM (Structural Similarity) [67]. Для квадратного окна a размера 8 × 8 на изображении I и соответствующего ему окна b на изображении I_w SSIM определяется как:

$$SSIM(a, b) = \frac{(2\mu_a\mu_b + c_1)(2\sigma_{ab} + c_2)}{(\mu_a^2 + \mu_b^2 + c_1)(\sigma_a^2 + \sigma_b^2 + c_2)}$$

где:

- μ_a, μ_b – средние значения a и b;

- σ_a^2, σ_b^2 – дисперсии a и b,
- σ_{ab} – ковариация a и b,
- $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ – переменные:
  - $L = 255$ – динамический диапазон пикселей,
  - $k_1 = 0.01$ и $k_2 = 0.03$ – константы.

Для вычисления метрик незаметности использовался набор из 40 объектов – изображений текстовых документов из открытых источников. В текстовый документ внедрялась бинарная последовательность: зафиксированная случайная последовательность и инверсия данной последовательности, последовательность из символов 0 и 1. При вычислении метрик незаметности сравнивались исходное изображение документа I и изображение документа с внедренной ЦВЗ I_w. Изображение I имеет размер $W \times H$ пикселей и динамический диапазон 8 бит на пиксель, или $I(x, y) \in \{0, \cdots, 255\}$, $x \in \{1, \cdots, W\}$, $y \in \{1, \cdots, H\}$. Если документ не позволял внедрить ЦВЗ выбранным методом, то документ не использовался при расчете метрики.

Таблица 5.1. Метрики визуальной заметности метода внедрения ЦВЗ в текстовые документы при печати.

Алгоритм	*PSNR*	*SSIM*	*Документов*
Горизонтальное смещение слов	16.5123	0.9009	**152**
Перечеркивание слов	**31.8115**	**0.9966**	124

Результаты вычисления метрик незаметности представлены в таблице 5.1. Согласно полученным результатам, значения метрик незаметности для метода на основе перечеркивания слов больше, чем у метода на основе горизонтального смещения слов. Однако, при экспертной оценке изображений с внедренными ЦВЗ было отмечено, что ЦВЗ на основе горизонтальных смещений слов более незаметен. ЦВЗ на основе

перечеркивания слов имитирует дефекты печати и визуально распознается экспертами. Данное противоречие объясняется особенностями человеческого восприятия, специфику которого метрики незаметности не учитывают полностью.

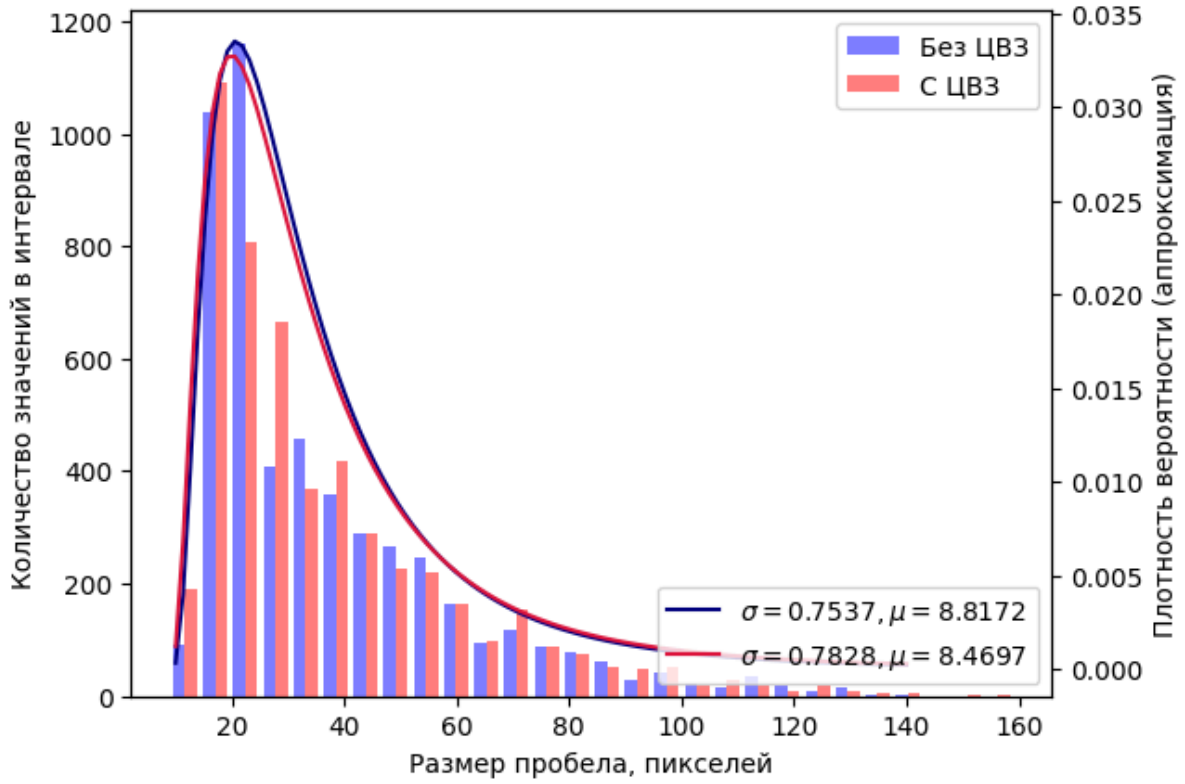


Рисунок 5.1. Распределение размеров пробелов в документах с и без ЦВЗ, аппроксимация логнормального распределения.

Стеганографический анализ – исследование на наличие скрытой информации в различных контейнерах информации (в данной работе в качестве контейнер выступает текстовый документ), таких как изображения, аудио и видео. Для метода внедрения ЦВЗ на основе горизонтального смещения слов была проведена оценка стойкости разработанного подхода к методу статистического стегоанализа в пространственной области. На рисунке 5.1 представлены гистограммы по размерам пробелов на изображениях документов исходных и содержащих ЦВЗ. Анализ статистических значений полученных величин пробелов между словами позволяет сделать вывод об отсутствии статистической

зависимости в распределении пробелов между словами как в оригинальных (без ЦВЗ), так и в изображениях текстовых документов с внедренным ЦВЗ. Визуальное сравнение гистограмм распределения величин интервалов между словами позволяет сделать вывод о подобии гистограммы текстового документа с ЦВЗ оригиналу.

5.1.3 Тестирование устойчивости ЦВЗ к искажениям

При печати документа исправным принтером возникает минимальное количество искажений, однако при обратной конвертации в цифровой формат возникают различные искажения:

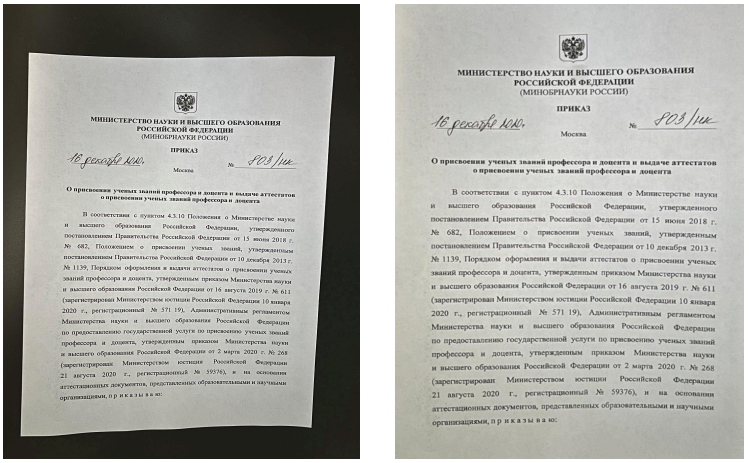
- *Уменьшение разрешения изображения* при сканировании документа с низким разрешением, при фотографировании с большого расстояния или при отправке изображения через приложения, выполняющие предобработку (например, мессенджеры WhatsApp или Telegram).
- *Перспективные искажения изображения* при фотографировании распечатанного документа, когда съемка осуществляется не под прямым углом к плоскости листа. Требуется восстановление исходной перспективы документа перед извлечением внедренной в ЦВЗ информации.
- *Искажение распределения цветов* при фотографировании с недостаточной или неравномерной освещенностью. Перед извлечением внедренной в ЦВЗ информации может потребоваться коррекция яркости, контраста или других параметров изображения документа.

Для оценки устойчивости разработанного метода внедрения ЦВЗ в текстовые документы разработана методика тестирования, приближенная к условиям реальной эксплуатации системы. Тестовый набор документов (40 документов) был сформирован на основе открытых источников и содержал

документы различного форматирования, изображения и таблицы. В текстовые документы внедрялась бинарная последовательность: зафиксированная случайная последовательность и инверсия данной последовательности, а также “нулевая” и “единичная” последовательности. Методика предполагает проверку работы алгоритмов внедрения и извлечения в сценариях:

- *Печать и сканирование (Print-Scan)* – в изображение текстового документа внедряется ЦВЗ, содержащий бинарную последовательность длины 32. Документ отправляется на печать и сканируется на устройстве KYOCERA TASKalfa 181, из сканированного изображения извлекается бинарная последовательность и сравнивается с внедренной;
- *Повторная печать и сканирование (Double-Print-Scan)* – в оцифрованное изображение распечатанного текстового документа с внедренным ЦВЗ повторно внедряется ЦВЗ, содержащий бинарную последовательность длины 32. Документ отправляется на печать и сканируется на устройстве KYOCERA TASKalfa 181, из сканированного изображения извлекается бинарная последовательность и сравнивается с внедренной;
- *Печать и фотографирование (Print-Cam)* – в изображение текстового документа внедряется ЦВЗ, содержащий бинарную последовательность длины 32. Документ отправляется на печать устройством KYOCERA TASKalfa 181, фотографируется на устройство Xiaomi Mi A1 с фиксированного расстояния 25 см и подвергается обработке для имитации отправки фотографии через мессенджер (изменение масштаба и сжатие алгоритмом JPEG с качеством 50). Из фотографии без обработки извлекается бинарная последовательность и сравнивается с внедренной; в другом сценарии фотографии подвергались *ручной обработке* (обрезка областей, не

относящихся к листу бумаги, цветокоррекция, удаление шумов и артефактов съемки) для повышения качества изображения документа и, следовательно, точности извлечения.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
(МИНОБРНАУКИ РОССИИ)

ПРИКАЗ

16 декабря 2020 г. № 903/нк

Москва

О присвоении ученых званий профессора и доцента и выдаче аттестатов о присвоении ученых званий профессора и доцента

В соответствии с пунктом 4.3.10 Положения о Министерстве науки и высшего образования Российской Федерации, утвержденного постановлением Правительства Российской Федерации от 15 июня 2018 г. № 682, Положением о присвоении ученых званий, утвержденным постановлением Правительства Российской Федерации от 10 декабря 2013 г. № 1139, Порядком оформления и выдачи аттестатов о присвоении ученых званий профессора и доцента, утвержденным приказом Министерства науки и высшего образования Российской Федерации от 16 августа 2019 г. № 611 (зарегистрирован Министерством юстиции Российской Федерации 10 января 2020 г., регистрационный № 571 19), Административным регламентом Министерства науки и высшего образования Российской Федерации по предоставлению государственной услуги по присвоению ученых званий профессора и доцента, утвержденным приказом Министерства науки и высшего образования Российской Федерации от 2 марта 2020 г. № 268 (зарегистрирован Министерством юстиции Российской Федерации 21 августа 2020 г., регистрационный № 59376), и на основании аттестационных документов, представленных образовательными и научными организациями, п р и к а з ы в а ю:

Рисунок 5.2. Пример фотографии документа до и после ручной обработки.

При тестировании метода внедрения ЦВЗ фиксировались следующие метрики:

- D – общее число документов при тестировании метода в определенном сценарии;
- I – количество изображений текстовых документов, в которые успешно внедрен ЦВЗ;
- E – количество изображений документов с внедренным ЦВЗ, из которых извлечена бинарная последовательность;
- $\overline{BER}$ – средняя доля ошибок в извлеченной бинарной последовательности;
- $\frac{E_{BER=0}}{I}$ – доля документов с внедренным ЦВЗ, из которых бинарная последовательность извлечена без ошибок.

Проведенные тесты (результаты в таблице 5.2) позволили сделать вывод о практической применимости метода. Метрика $\frac{E_{BER=0}}{I}$ отражает долю деанонимизированных утечек текстовых документов с внедренной

ЦВЗ. Метод на основе горизонтального смещения показал значение метрики $\frac{E_{BER=0}}{I}$ в 61.7% для сканированных документов, 56.7% для сканированных документов, в которые ЦВЗ был внедрен повторно, а также 56.5% для фотографий. Ручная предобработка фотографий позволила увеличить долю деанонимизированных утечек до 69.7%. Метод на основе перечеркивания позволил деанонимизировать более 80% утечек документов во всех сценариях: 84% для сканированных документов, 82.5% для сканированных документов, в которые ЦВЗ была внедрена повторно, и 80.3% для фотографий.

Таблица 5.2. Результаты тестирования метода внедрения ЦВЗ на основе различных структурных подходов кодирования информации.

Алгоритм	*Сценарий*	$\overline{BER}$	$\frac{E_{BER=0}}{I}$	*E*	*I*	*D*
Горизонтальное смещение слов	*Print-Scan*	0.144	0.617	127	136	160
	Double-Print-Scan	0.189	0.567	128	134	160
	Print-Cam	0.185	0.565	74	76	80
	Print-Cam + ручная обработка	0.123	0.697	74	76	80
Перечеркивание слов	*Print-Scan*	0.001	0.840	112	132	160
	Double-Print-Scan	0.003	0.825	112	132	160
	Print-Cam	0.006	0.803	57	66	80

5.2 Тестирование метода внедрения ЦВЗ при выводе на экран

Метод внедрения ЦВЗ при выводе текстовых документов на экран должен быть устойчив к различным искажениям, возникающим при выводе на экран, независим к различиям в характеристиках мониторов и фотокамер, а также условиям фотографирования экрана. Искажения можно разделить на две группы:

- *Искажения, связанные с процессом вывода изображения на экран*

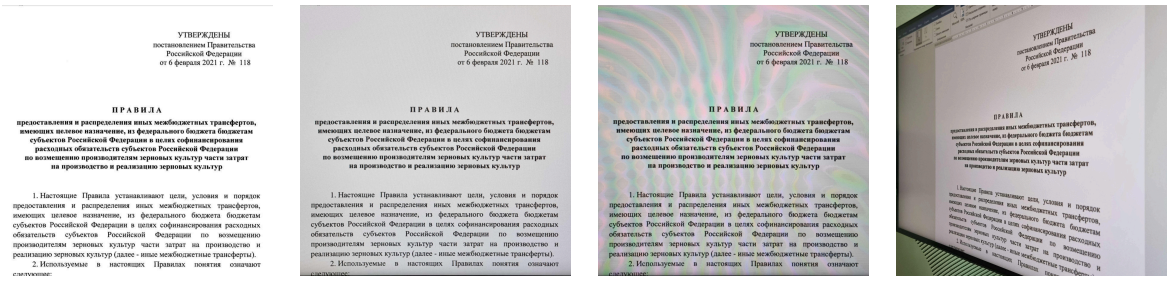
  Включают в себя преобразование цифрового сигнала в аналоговый,

которое может сильно варьироваться между разными мониторами. Эти различия могут влиять на цвет и яркость пикселей изображения, видимые человеческим глазом или цифровой камерой.

- *Искажения, связанные с фотографированием экрана и условиями съемки*

  Включают в себя расстояние и угол между камерой и экраном, что может изменять исходную форму и масштаб изображения. Оптическая система камеры может также вызывать различные оптические эффекты, такие как бочкообразная дисторсия. Фотография также зависит от наличия дополнительных источников света и настройки фокусировки камеры.

Описанный в Главе 4 метод внедрения ЦВЗ был реализован и протестирован. Обучены нейронные сети E, D_c и D_w, емкость ЦВЗ составляет $M = 50$ бит при размере изображения I_w, равном $S = 120$. Для оптимизации параметров нейронных сетей был использован алгоритм *Adam* [39]. В общей сложности в процессе обучения было сгенерировано 1440000 50-битных последовательностей. Обучение проводилось с использованием графического адаптера Nvidia RTX 3060 и заняло около 8 часов.



Оригинальный документ | *Фотография экрана* | *Фотография экрана с эффектом муара* | *Фотография экрана с искаженной перспективой*

Рисунок 5.3. Фотографии документа с различными искажениями.

Реализация предполагает встраивание 32-битного идентификатора сотрудника и устройства, дополненного 18 битами БЧХ-кода, позволяющих обнаруживать и исправлять до 3 ошибок в полученной 50-битной последовательности. Считается, что бинарная последовательность извлечена корректно, если в извлеченной 50-битной последовательности не более 3 ошибок.

При тестирования метода внедрения ЦВЗ были сделаны фотографии экранов с выведенным текстовыми документами. В качестве источников документов были использованы открытые источники: сайт министерства образования и науки РФ [13] и сайт министерства финансов РФ [15]. В тестировании принимали участие 3 монитора, характеристики которых приведены в таблице 5.3, и 3 смартфона, характеристики цифровых камер которых приведены в таблице 5.4. Съемка проводилась при помощи встроенного приложения производителя смартфона на основную камеру в автоматическом режиме.

Таблица 5.3. Характеристики мониторов, использованных при тестировании метода внедрения ЦВЗ.

Модель монитора	*Тип матрицы*	*Разрешение экрана*	*Частота развертки*
Samsung SyncMaster 940FN	TFT PVA	1280 × 1024	75 Гц
Sony SDM-S75A	TN	1280 × 1024	75 Гц
Dell U2722D	IPS	2560 × 1440	60 Гц

Таблица 5.4. Характеристики камер, использованных при тестировании метода внедрения ЦВЗ.

Модель смартфона	*Разрешение*	*Апертура*	*Фокусное расстояние*	*Размер сенсора*
Xiaomi Mi A1	12 Мп	f/2.2	26 мм	1.25 мкм
Samsung Galaxy S8	12 Мп	f/1.7	26 мм	1.4 мкм
Samsung Galaxy S21	12 Мп	f/1.8	26 мм	1.8 мкм

5.2.1 Подбор коэффициента непрозрачности

Для определения степени незаметности использовались метрики PSNR и SSIM (определены в разделе 5.1.2), широко применяемые в задаче сжатия изображений. Сравнивались исходное изображение I и изображение с ЦВЗ I_w. Монохромное изображение I имеет размер $W \times H$ пикселей и динамический диапазон 8 бит на пиксель, или $I(x, y) \in \{0, \cdots, 255\}$, $x \in \{1, \cdots, W\}$, $y \in \{1, \cdots, H\}$. Для изображений $SSIM(I, I_w)$ вычисляется усреднением по набору окон a и b. Значение 1 соответствует полному совпадению сравниваемых изображений. Изображения можно считать схожими при значениях SSIM выше 0.97.

На незаметность ЦВЗ на экране влияет значение коэффициента непрозрачности окна-оверлея α. Исходя из особенностей реализации, значения α удобно представлять в дробях вида $\frac{n}{255}$, где n — целое число. При расчете проводилось усреднение по изображениям 50 документов. Поскольку ЦВЗ отображается вне зависимости от выводимого на экран изображения, важно добиться его незаметности не только на изображениях документов, но и на произвольных изображениях. Поэтому значения метрик оценки заметности были посчитаны также для 50 цветных изображений, взятых из набора Open Images V6 [40]. Результаты расчетов метрик заметности для различных α представлены в таблице 5.5.

Таблица 5.5. Значения метрик незаметности ЦВЗ.

α	*Изображения документов*		*Цветные изображения*	
	PSNR, дБ	*SSIM*	*PSNR, дБ*	*SSIM*
3/255	44.4	0.9992	47.5	0.9972
4/255	42.6	0.9991	45.5	0.9958
5/255	40.6	0.9989	43.8	0.9942
6/255	39.0	0.9987	42.4	0.9926
7/255	37.7	0.9985	41.1	0.9910

8/255	36.5	0.9983	40.0	0.9894
9/255	35.4	0.9980	39.0	0.9878
10/255	34.5	0.9977	38.1	0.9863

Увеличение коэффициента непрозрачности α окна-оверлея приводит к повышению заметности ЦВЗ. Согласно значениям метрики SSIM метод демонстрирует высокую незаметность при всех значениях α, использованных при тестировании. Метрика PSNR показывает, что встраиваемый ЦВЗ менее заметен на цветных изображениях, чем на изображениях документов. При значениях α, не превосходящих $5/255$, ЦВЗ мало заметен на всех типах рассмотренных изображений.

Для определения «рабочего» значения непрозрачности α необходимо также проверить точность извлечения внедренной в ЦВЗ информации. Для этого было проведено тестирование для 9 пар камер и мониторов в заданном диапазоне значений α. Съемка проводилась на расстоянии 40 сантиметров при параллельном расположении плоскостей камеры и экрана. Для каждой пары и каждого значения непрозрачности было сделано 10 фотографий документов различного масштаба, отображаемых на экране.

Таблица 5.6. Точность извлечения внедренной в ЦВЗ информации при различных значениях незаметности.

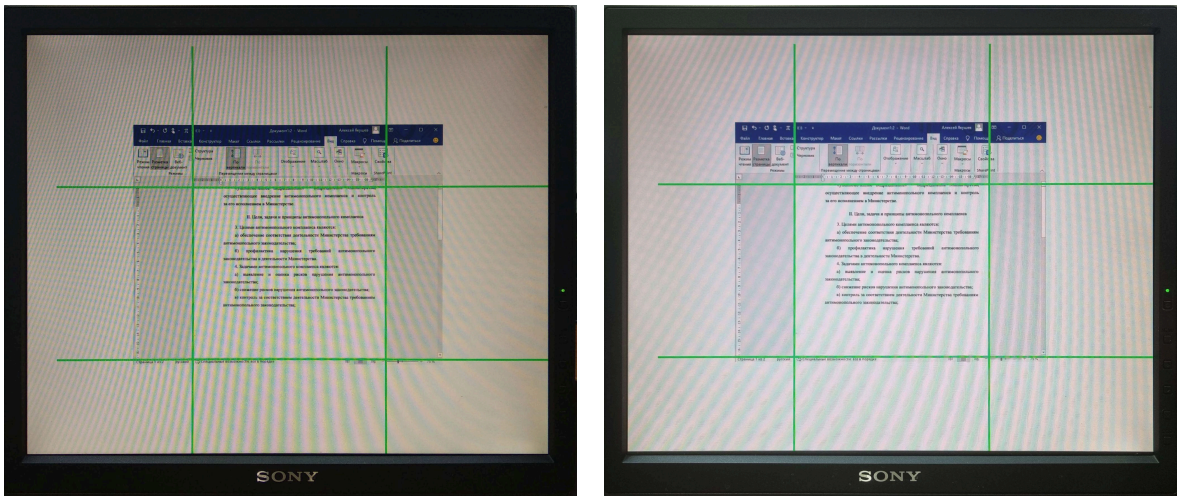
α	*Камера*					
	Xiaomi Mi A1		*Samsung Galaxy S8*		*Samsung Galaxy S21*	
	BER	$\leq$ 3 ош.	BER	$\leq$ 3 ош.	BER	$\leq$ 3 ош.
Монитор Samsung SyncMaster 940FN						
3/255	33.6%	3/10	18.4%	7/10	17.6%	6/10
4/255	9.1%	7/10	1.4%	10/10	6.8%	9/10
5/255	10.6%	7/10	0.6%	10/10	0.2%	10/10
6/255	1.0%	10/10	0.4%	10/10	0.2%	10/10
7/255	0.6%	10/10	0.0%	10/10	0.2%	10/10

8/255	0.2%	10/10	0.0%	10/10	0.2%	10/10
9/255	0.2%	10/10	0.0%	10/10	0.0%	10/10
10/255	0.0%	10/10	0.0%	10/10	0.0%	10/10
Монитор Sony SDM-S75A						
3/255	44.8%	0/10	50.2%	0/10	48.0%	1/10
4/255	48.4%	0/10	14.6%	8/10	36.8%	3/10
5/255	41.0%	1/10	2.0%	10/10	34.6%	3/10
6/255	41.4%	2/10	1.4%	10/10	31.6%	4/10
7/255	32.2%	4/10	1.4%	10/10	21.6%	5/10
8/255	6.0%	9/10	0.8%	10/10	20.4%	6/10
9/255	6.4%	9/10	0.4%	10/10	16.8%	7/10
10/255	4.0%	9/10	0.6%	10/10	11.2%	8/10
Монитор Dell U2722D						
3/255	7.8%	8/10	8.8%	8/10	14.8%	7/10
4/255	1.8%	9/10	1.4%	10/10	1.2%	10/10
5/255	1.0%	10/10	1.0%	10/10	1.0%	10/10
6/255	0.8%	10/10	0.6%	10/10	0.4%	10/10
7/255	0.2%	10/10	0.2%	10/10	0.0%	10/10
8/255	0.0%	10/10	0.2%	10/10	0.0%	10/10
9/255	0.0%	10/10	0.0%	10/10	0.0%	10/10
10/255	0.0%	10/10	0.2%	10/10	0.0%	10/10

В таблице отображена доля бит, извлеченных с ошибками BER (Bit Error Rate), а также число фотографий, из которых внедренная бинарная последовательность была извлечена с 3 (максимальное число корректируемых при помощи БЧХ-кода ошибок инверсии) или менее ошибками. Если значение BER равно 0, внедренная информация извлечена из всех фотографий полностью корректно, если же оно близко к 0.5, извлечение из всех фотографий прошло неудачно. Стоит отметить, что разработанный метод осуществляет наложение ЦВЗ на экран вне зависимости от содержимого выводимого на экран документа, в отличии

от структурного метода встраивания ЦВЗ, предъявляющего требования к объему и структуре текста в документе.

При $\alpha > 7/255$ внедренная бинарная последовательность корректно извлекается практически из всех фотографий, за исключением фотографий экрана монитора Sony SDM-S75A, сделанных на камеры смартфонов Xiaomi Mi A1 и Samsung Galaxy S21. На этих фотографиях достаточно сильно проявляется эффект муара (рисунок 5.4). В дальнейших экспериментах значения непрозрачности α были зафиксированы для каждого монитора: $7/255$ для монитора Samsung SyncMaster 940FN, $8/255$ для монитора Sony SDM-S75A, $6/255$ для монитора Dell U2722D. Данные мониторы по-разному отображают одинаковые изображения, различные значения коэффициента непрозрачности α выбраны для одинаковой перцептивной заметности ЦВЗ.



(a) Xiaomi Mi A1 (b) Samsung Galaxy S21

Рисунок 5.4. Примеры фотографий монитора Sony SDM-S75A на расстоянии 40 см.

5.2.2 Определение зависимости точности извлечения от расстояния между камерой и экраном

Для всех пар мониторов и камер были сделаны фотографии на различных расстояниях между камерой и экраном при параллельном

расположении плоскости камеры относительно плоскости экрана. Произведена попытка извлечь ЦВЗ из полученных фотографий.

Таблица 5.7. Влияние расстояния от камеры до экрана на точность извлечения внедренной в ЦВЗ информации.

Расстояние между камерой и экраном	*Камера*					
	Xiaomi Mi A1		*Samsung Galaxy S8*		*Samsung Galaxy S21*	
	BER	$\leq$ 3 ош.	*BER*	$\leq$ 3 ош.	*BER*	$\leq$ 3 ош.
Монитор Samsung SyncMaster 940FN						
25 см	1.6%	10/10	45.0%	0/10	1.4%	10/10
40 см	0.4%	10/10	0.2%	10/10	0.8%	10/10
60 см	0.0%	10/10	0.8%	10/10	0.0%	10/10
80 см	0.6%	10/10	1.2%	10/10	0.2%	10/10
100 см	0.4%	10/10	1.2%	9/10	0.6%	10/10
Монитор Sony SDM-S75A						
25 см	34.0%	3/10	4.8%	8/10	22.4%	6/10
40 см	32.0%	4/10	0.8%	10/10	22.0%	6/10
60 см	0.6%	10/10	0.6%	10/10	0.4%	10/10
80 см	0.4%	10/10	1.2%	10/10	0.4%	10/10
100 см	0.8%	10/10	2.0%	10/10	0.2%	10/10
Монитор Dell U2722D						
25 см	27.8%	4/10	9.6%	5/10	30.8%	4/10
40 см	1.0%	10/10	1.2%	10/10	0.8%	10/10
60 см	0.4%	10/10	7.0%	9/10	18.6%	6/10
80 см	0.2%	10/10	4.4%	9/10	0.2%	10/10
100 см	0.0%	10/10	4.4%	9/10	0.6%	10/10

Как и в предыдущем эксперименте, наблюдается высокая доля ошибок при извлечении внедренной в ЦВЗ информации из фотографий экрана монитора Sony SDM-S75A, сделанных на камеры смартфонов Xiaomi Mi A1 и Samsung Galaxy S21 на расстоянии 40 см. При этом точность извлечения из фотографий, сделанных на других расстояниях

значительно выше. Было сделано предположение, что данная особенность связана с усилением эффекта муара на фотографии, возникающим при определенных условиях съемки, одним из которых является расстояние между камерой и экраном. Также на многих парах мониторов и камер эффект муара проявляется на расстоянии 25 см.

5.2.3 Определение зависимости точности извлечения от угла между камерой и экраном

Для проверки влияния искажения перспективы фотографии на точность извлечения ЦВЗ были сделаны фотографии под различными углами между плоскостью камеры и плоскостью экрана. При этом расстояние между камерой и центром экрана было зафиксировано и составляло 40 см. Результаты эксперимента представлены в таблице 5.8.

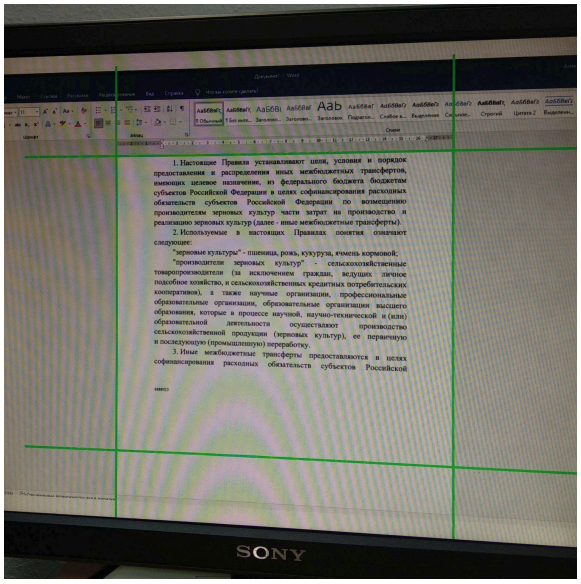
Таблица 5.8. Точность извлечения внедренной в ЦВЗ информации из фотографий, сделанных под углом к экрану.

Угол между камерой и экраном	*Камера*					
	Xiaomi Mi A1		*Samsung Galaxy S8*		*Samsung Galaxy S21*	
	BER	$\leq$ 3 ош.	*BER*	$\leq$ 3 ош.	*BER*	$\leq$ 3 ош.
Монитор Samsung SyncMaster 940FN						
0°	0.4%	10/10	0.2%	10/10	0.8%	10/10
15°	0.8%	10/10	0.4%	10/10	1.2%	10/10
30°	14.6%	6/10	1.0%	10/10	1.6%	9/10
45°	0.4%	9/10	1.2%	10/10	0.2%	10/10
60°	1.2%	10/10	1.4%	10/10	1.8%	10/10
Монитор Sony SDM-S75A						
0°	32.0%	4/10	0.8%	10/10	22.0%	6/10
15°	8.4%	8/10	1.6%	10/10	15.0%	7/10
30°	2.0%	10/10	1.8%	10/10	1.4%	10/10
45°	2.4%	10/10	1.4%	10/10	1.4%	10/10
60°	1.4%	10/10	1.6%	10/10	1.0%	8/10

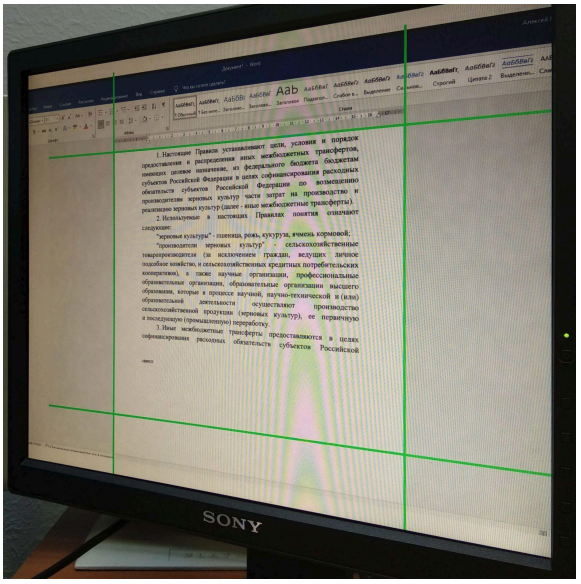
Монитор Dell U2722D						
0°	1.0%	10/10	1.2%	10/10	0.8%	10/10
15°	6.6%	8/10	7.4%	9/10	1.4%	10/10
30°	1.4%	10/10	7.4%	9/10	1.8%	10/10
45°	1.8%	10/10	3.8%	9/10	1.4%	10/10
60°	1.6%	9/10	6.8%	9/10	1.2%	10/10

При извлечении внедренной в ЦВЗ информации из фотографий экрана монитора Sony SDM-S75A, сделанных на камеры смартфонов Xiaomi Mi A1 и Samsung Galaxy S21, наблюдалась следующая особенность: на фотографиях с большим искажением перспективы эффект муара менее выражен (рисунок 5.5), благодаря чему точность извлечения ЦВЗ оказалась выше.

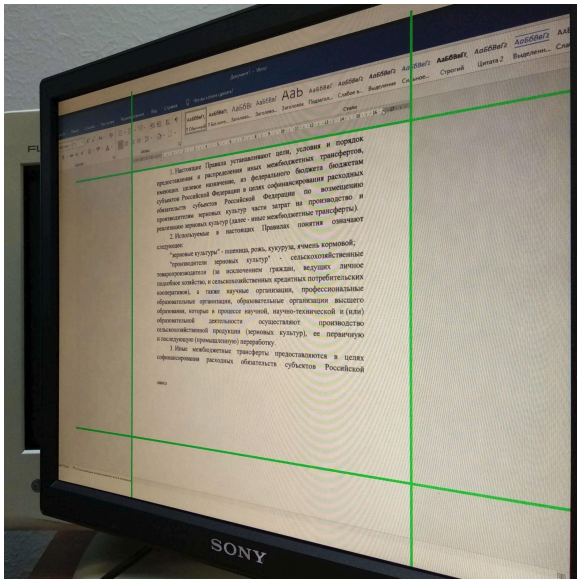
Эффект муара также проявляется на фотографиях экрана монитора Samsung SyncMaster 940FN, сделанных на камеру смартфона Xiaomi Mi A1 под углами 15° и 30°, что повлияло на точность извлечения ЦВЗ из этих фотографий.



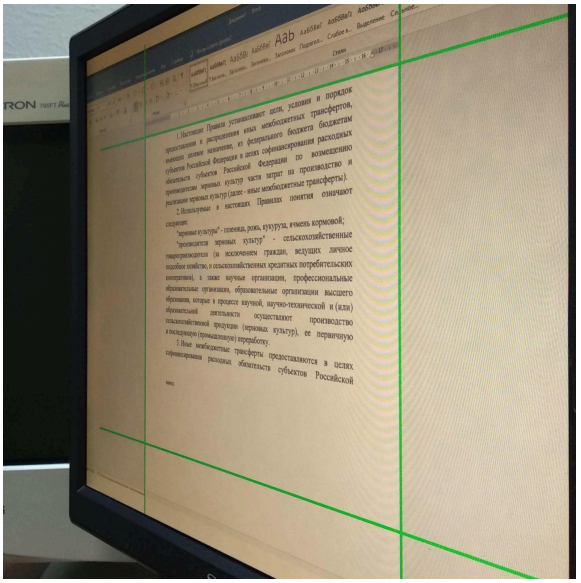
(a) 15°



(b) 30°



(c) 45°



(d) 60°

Рисунок 5.5. Примеры фотографий монитора Sony SDM-S75A на расстоянии 40 см.

5.2.4 Определение зависимости точности извлечения от степени сжатия JPEG

Предложенный метод наложения ЦВЗ на выводимые на экран документы был протестирован на устойчивость к сжатию фотографий посредством алгоритма JPEG. Из предыдущих экспериментов случайным образом было отобрано 50 фотографий, из которых внедренная бинарная последовательность была извлечена с 3 или менее ошибками. Фотографии подвергались сжатию алгоритмом JPEG с различными значениями коэффициента качества. Значения точности извлечения из сжатых фотографий занесены в таблицу 5.9.

Таблица 5.9. Влияние степени сжатия фотографий на точность извлечения внедренной в ЦВЗ информации.

Коэффициент сжатия JPEG	*Средний размер файла, Кб*	*BER*	≤ 3 *ошибок*
Без сжатия	4589	1.3%	50/50
80	1870	1.2%	50/50
60	1211	1.2%	50/50

50	1048	1.4%	49/50
40	900	1.4%	49/50
30	748	4.4%	46/50
20	573	8.5%	42/50
15	478	14.5%	34/50
10	376	26.9%	20/50

Внедренная в ЦВЗ информация полностью извлекается из фотографий после сжатия по алгоритму JPEG с коэффициентом качества не менее 50, а также частично извлекается при значении коэффициента качества не ниже 20.

5.2.5 Оценка результатов тестирования метода внедрения ЦВЗ в документы при выводе на экран

Метод внедрения ЦВЗ в текстовые документы обеспечивает внедрение 32-битного идентификатора сотрудника и устройства в текстовые документы при выводе на экран. При значении коэффициента непрозрачности $\alpha = 7/255$ внедренный в ЦВЗ идентификатор извлечен без ошибок из 86.67% тестовых фотографий. Незаметность ЦВЗ на данном уровне непрозрачности оценивается в 37.7 дБ по метрике PSNR и 0.9985 по метрике SSIM. Снижение α приводит к уменьшению доли успешных извлечений, но повышает незаметность ЦВЗ. Увеличение α до $8/255$ привело к повышению доли успешных извлечений ЦВЗ до 94.44%, и при этом понизило незаметность до 36.5 дБ (PSNR) и 0.9983 (SSIM).

Оценка влияния расстояния и угла съемки на точность извлечения показала, что эффект муара приводит к возникновению ошибок при извлечении внедренной в ЦВЗ информации. При фотографировании экрана на расстоянии ≥40 см и с углом съемки ≤60° доля успешных извлечений превышает 87.78%. Тестирование устойчивости к JPEG-сжатию показало, что ЦВЗ корректно извлекается из 96%

фотографий при качестве JPEG не менее 40 и падает до 84% при качестве JPEG 20, используемом в мессенджере WhatsApp при пересылке изображения.

5.3 Выводы

В пятой главе представлены методика и результаты тестирования системы противодействия анонимности при утечках текстовых документов. Протестированы методы, обеспечивающие внедрение 32-битного идентификатора сотрудника и устройства с помощью ЦВЗ в текстовые документы при печати и выводе на экран.

Метод внедрения ЦВЗ при печати на основе горизонтального смещения позволил деанонимизировать 61.7% утечек сканированных документов и 56.5% фотографий (69.7% с ручной предобработкой), метод на основе перечеркивания слов позволил деанонимизировать более 80% утечек документов во всех сценариях. При выводе документов на экран ЦВЗ накладывается с помощью окна-оверлея с непрозрачностью $\alpha = 7/255$, обеспечивая 86.67% успешных извлечений. Увеличение непрозрачности до 8/255 повышает точность извлечения до 94.44%, но также снижает визуальную незаметность водяного знака. Метод наложения ЦВЗ на экран показал высокую устойчивость к фотографированию под разными углами и с различного расстояния, а также к сжатию фотографий, позволив деанонимизировать более 80% утечек в большинстве сценариев. По итогу, система противодействия анонимности при утечке позволяет идентифицировать ее источник по изображению фотографии экрана, фотографии или сканированному изображению распечатанного документа более, чем в 80% случаев.

Заключение

Основные результаты работы:

1. Разработана архитектура системы деанонимизации при утечках изображений текстовых документов, обеспечивающая внедрение уникальных идентификаторов сотрудников и используемых ими устройств в документы при печати и выводе на экран;
2. Разработан обладающий научной новизной структурный метод внедрения ЦВЗ, предполагающий слепое извлечение внедренной информации, на основе сегментации изображения документа с помощью нейросетевого алгоритма. Разработанный метод обладает визуальной незаметностью и устойчивостью к искажениям, возникающим при печати и последующей оцифровке посредством фотографирования или сканирования, и ориентирован для выполнения на процессоре общего назначения с минимальным потреблением вычислительных ресурсов;
3. Разработан обладающий научной новизной метод генерации ЦВЗ нейросетевым алгоритмом, предполагающий слепое извлечение внедренной информации и обладающий свойствами визуальной незаметности и устойчивости к искажениям, возникающим при фотографировании экрана и сжатии алгоритмами, применяемыми в мессенджерах при пересылке изображений;
4. На основе разработанных архитектуры и методов внедрения/извлечения ЦВЗ реализована система противодействия анонимности при утечках текстовых документов. Система протестирована на целевых сценариях утечек – фотографирование документов на экране и сканирование распечатанных документов. Апробация подтвердила эффективность системы в различных

условиях и продемонстрировала способность успешно извлекать внедренную в ЦВЗ информацию и деанонимизировать утечки.

Также определены направления дальнейшей работы:

- повышение точности и устойчивости к искажениям разработанных методов внедрения ЦВЗ в текстовые документы;
- разработка методов внедрения ЦВЗ в изображения, отличные от документов, а также аудио- и видеоконтент;
- исследование применимости технологии водяных знаков для решения других практических задач

Список литературы

1. Варшамов Р. Р., Тененгольц Г. М. Код, исправляющий одиночные несимметрические ошибки // Автоматика и телемеханика. — 1965. — Т. 26, № 2. — С. 288-292.
2. Гетьман А. И., Обыденков Д. О., Фролов А. Е., Маркин Ю. В. Методы маркирования текстовых документов при печати // Ежегодная научная конференция «Ломоносовские чтения», секция «Вычислительной математики и кибернетики». — 2021. — Р. 55–57.
3. Козачок А. В., Козачок В. И., Копылов С. А., Горбачев П. Н., Маркин Ю. В., Обыденков Д. О. Экспериментальная оценка алгоритма маркирования текстовых документов на основе изменения интервалов между словами // Труды Института системного программирования РАН. — 2022. — Т. 34, № 4. — С. 153–172.
4. Козачок А. В., Копылов С. А., Горбачев П. Н., Гайнов А. Е., Кондратьев Б. В. Алгоритм маркирования текстовых документов на основе изменения интервалов между словами, обеспечивающий устойчивость к преобразованию формата // Труды Института системного программирования РАН. — 2021. — Т. 33. — № 4. — С. 131–146.
5. Козлов С. В., Копылов С. А., Кондратьев Б. В., Обыденков Д. О. Реализация маркирования в подсистеме печати ОС семейства Windows на основе виртуального XPS-принтера // Труды Института системного программирования РАН. — 2020. — Т. 32, № 5. — С. 95–110.
6. Национальный стандарт Российской Федерации. Система стандартов по информации, библиотечному и издательскому делу. Организационно-распорядительная документация. Требования к оформлению документов. ГОСТ Р 7.0.97–2016. — М.: Стандартинформ, 2019. — 32 с. (на русском языке).
7. Обыденков Д. О., Фролов А. Е., Маркин Ю. В., Фомин С. А., Кондратьев Б. В. Методы маркирования текстовых документов при печати посредством вертикального сдвига и изменения яркости фрагментов слов // Труды Института системного программирования РАН. — 2021. — Т. 33. — № 5. — С. 65–82.

8. Обыденков Д. О., Якушев А. Ю., Маркин Ю. В., Фомин С. А., Фролов А. Е., Козлов С. В., Громей Д. Д., Козачок А. В., Кондратьев Б. В. Система маркирования документов для проведения расследований при их утечке // Труды Института системного программирования РАН. — 2021. — Т. 33, № 6. — С. 161–174.
9. Обыденков Д. О., Якушев А. Ю., Фомин С. А., Маркин Ю. В., Козачок А. В., Фролов А. Е., Козлов С. В., Громей Д. Д., Акименков А. А., Мякутин А. В., Челина В. А. Предотвращение анонимных утечек конфиденциальных документов // Материалы 33-й Научно-технической конференции «Методы и технические средства обеспечения безопасности информации». — 2024. — С. 114–115.
10. Писковский В.О., Грушо А.А. Программный комплекс Абонентский облачный терминал // XIV Всероссийское совещание по проблемам управления (ВСПУ-2024) : сборник научных трудов, 17-20 июня 2024 г., Москва / Под общ. ред. Д.А. Новикова. — Электрон. текстовые дан. (824 файла: 433 МБ). — М.: ИПУ РАН, 2024. — С. 2935–2939.
11. Писковский В.О., Семинихин Д.А., Грушо А.А., Забежайло М.И. Метод идентификации рабочего места по фотоснимку экрана компьютера // Вестник Московского университета. Серия 15. Вычислительная математика и кибернетика. — 2023. — № 3. — С. 56–72.
12. Программный комплекс ВИКОНТ. Противодействие краже информации [Электронный ресурс]. URL: https://12ikont.ru/ (дата обращения: 11.10.2024).
13. Список документов на сайте министерства науки и высшего образования Российской Федерации [Электронный ресурс]. URL: https://minobrnauki.gov.ru/documents/ (дата обращения: 11.10.2024).
14. Семенихин Д.А., Писковский В.О. Разработка системы идентификации канала утечки информации по фотографии документа, сделанной с экрана компьютера // Ломоносовские чтения-2020. Секция «Вычислительной математики и кибернетики». — М.: Изд-во Моск. ун-та, 2020. — С. 128.
15. Список документов на сайте министерства финансов Российской Федерации [Электронный ресурс]. URL:

https://minfin.gov.ru/ru/document/ (дата обращения: 11.10.2024).

16. Уоррен, С. Г., мл. Алгоритмические трюки для программистов = Hacker’s Delight. — М.: Вильямс, 2007. — 288 с. — ISBN 0-201-91465-4.
17. Якушев А. Ю., Маркин Ю. В., Фомин С. А., Обыденков Д. О., Кондратьев Б. В. Маркирование текстовых документов на экране монитора посредством изменения яркости фона в областях межстрочных интервалов // Труды Института системного программирования РАН. — 2021. — Т. 33, № 4. — С. 147–162.
18. A Novel Arabic Text Steganography Method Using Letter Points and Extensions // In: WASET International Conference on Computer, Information and Systems Science and Engineering (ICCISSE), May 25-27, 2007, Vienna, Austria. — 2007.
19. Alattar A. M., Alattar O. M. Watermarking electronic text documents containing justified paragraphs and irregular line spacing // Security, Steganography, and Watermarking of Multimedia Contents VI. — 2004. — Т. 5306. — С. 685–695.
20. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H. Character region awareness for text detection. — 2019. — С. 9365–9374. — В сборнике: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.
21. Bose R. C., Ray-Chaudhuri D. K. On a class of error correcting binary group codes // Information and Control. — 1960. — Т. 3, № 1. — С. 68-79.
22. Brassil J. T., Low S. Electronic marking and identification techniques to discourage document copying // IEEE Journal on Selected Areas in Communications. — 1995. — Т. 13, № 8. — С. 1495–1504.
23. Ch'ng, C. K., Chan, C. S. Total-text: A comprehensive dataset for scene text detection and recognition. — 2017. — Т. 1. — С. 935–942. — В сборнике: 2017 14th IAPR international conference on document analysis and recognition (ICDAR). — IEEE.
24. Chen W., Li Y., Niu Z., Xu Y., Keskinarkaus A., Seppänen T., Sun X. Real-time and screen-cam robust screen watermarking // Knowledge-Based Systems. — 2024. — Т. 302. — С. 112380.

25. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition. — 2009. — С. 248-255. — IEEE.
26. Docs Security Suite [Электронный ресурс]. URL: https://ct-sg.ru/products/docs-security-suite/ (дата обращения: 11.10.2024).
27. Dong P., Brankov J. G., Galatsanos N. P., Yang Y., Davoine F. Digital watermarking robust to geometric distortions // IEEE Transactions on Image Processing. — 2005. — Т. 14, № 12. — С. 2140—2150.
28. EveryTag [Электронный ресурс]. URL: https://everytag.ru/ (дата обращения: 11.10.2024).
29. IEEE Standard for Floating-Point Arithmetic. Revision of IEEE Std 754—2008 // IEEE. — 2019. — ISBN 978-1-5044-5924-2, doi:10.1109/IEEESTD.2019.8766229.
30. Forbes. 15 крупнейших российских работодателей с числом сотрудников более 100 000 человек [Электронный ресурс]. — URL: https://www.forbes.ru/biznes/503283-15-krupnejsih-rossijskih-rabotodatelej-s-cislom-sotrudnikov-bolee-100-000-celovek (дата обращения: 12.10.2024).
31. Ge S., Fei J., Xia Z., Tong Y., Weng J., Liu J. A screen‐shooting resilient document image watermarking scheme using deep neural network // IET Image Processing. — 2023. — Т. 17, № 2. — С. 323-336.
32. Gtk::Window::set_keep_above [Электронный ресурс]. URL: https://docs.gtk.org/gtk3/method.Window.set_keep_above.html (дата обращения: 11.10.2024).
33. Gugelmann D., Sommer D., Lenders V., Happe M., Vanbever L. Screen watermarking for data theft investigation and attribution // In: 2018 10th International Conference on Cyber Conflict (CyCon). — IEEE, 2018. — С. 391-408.
34. Grusho A., Piskovski V., Semenikhin D., Sudarikov I., Timonina E. The research of a method to identify a workplace via a monitor snapshot // 2020 International Scientific and Technical Conference Modern Computer Network Technologies (MoNeTeC). — IEEE, 2020. — P.

1–6.

35. Hamming R. W. Error detecting and error correcting codes // The Bell System Technical Journal. — 1950. — Т. 29, № 2. — С. 147-160.
36. Handwritten Signatures Dataset [Электронный ресурс]. 2018. URL: https://www.kaggle.com/datasets/divyanshrai/handwritten-signatures (дата обращения: 11.10.2024).
38. Kim Y.W., Moon K.A., Oh I. S. A Text Watermarking Algorithm based on Word Classification and Interword Space Statistics // Proc. of the Seventh International Conference on Document Analysis and Recognition. — 2003. — С. 775–779.
39. Kingma, Д. П. Adam: метод стохастической оптимизации // arXiv preprint arXiv:1412.6980 — 2014.
40. Kuznetsova A., Rom H., Alldrin N. et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale // International Journal of Computer Vision. — 2020. — Vol. 128, no. 7. — P. 1956–1981.
41. Low S. H., Maxemchuk N. F., Brassil J. T., O'Gorman L. Document marking and identification using both line and word shifting // Proceedings of INFOCOM'95. — Т. 2. — IEEE, 1995. — С. 853—860.
42. McAfee Data Loss Prevention [Электронный ресурс]. URL: https://docs.mcafee.com/bundle/data-loss-prevention-11.4.x-product-guide (дата обращения: 24.10.2021).
43. Porter T., Duff T. Compositing digital images // Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques. — 1984. — P. 253–259.
44. Pramila A. Reading watermarks with a camera phone from printed images. — Оулу: Университет Оулу, 2018. — 13 февр. 2018. — Диссертация на соискание ученой степени доктора философии.
45. PRINTER GUARD. Система контроля и экономии ресурсов печати [Электронный ресурс]. URL: https://secretgroup.ru/nashi-produkty/printer-guard/ (дата обращения: 11.10.2024).
46. Rec I. BT 601: Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios // ITU-R Rec. BT. —

1995. — Vol. 656.

47. Reed I. S., Solomon G. Polynomial codes over certain finite fields // Journal of the Society for Industrial and Applied Mathematics. — 1960. — Т. 8, № 2. — С. 300-304.

48. Retributor [Электронный ресурс]. URL: https://retributor.ru/ (дата обращения: 11.10.2024).

49. Ronneberger, O., Fischer, P., Brox, T. U-net: Convolutional networks for biomedical image segmentation. — 2015. — С. 234–241. — В сборнике: Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III. — Springer.

50. SafeCopy. Решение для защиты печатных, графических и электронных копий документов [Электронный ресурс]. URL: https://www.evraas.ru/upload/iblock/652/6522428b044b73f3b7ad2e5c58ba292b.pdf (дата обращения: 11.10.2024).

51. Sara U., Akter M., Uddin M. S. Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study // Journal of Computer and Communications. — 2019. — Vol. 7, No 3. — P. 8–18.

52. SCREEN GUARD. Система снижения рисков утечки информации путем фотографирования [Электронный ресурс]. URL: https://secretgroup.ru/nashi-produkty/screen-guard/ (дата обращения: 11.10.2024).

53. Shirali-Shahreza M. A. A New Persian/Arabic Text Steganography Using “La” Word // In: Advances in Computer and Information Sciences and Engineering. — Springer: Berlin/Heidelberg, Germany, 2008. — С. 339–342.

54. Shirali-Shahreza M. H., Shirali-Shahreza M. A new approach to Persian/Arabic text steganography // 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (ICIS-COMSAR’06). — IEEE, 2006. — С. 310—315.

55. Smith, R. An overview of the Tesseract OCR engine. — 2007. — Т. 2. — С. 629–633. — В сборнике: Ninth international conference on document analysis and recognition (ICDAR 2007). — IEEE.

56. Symantec Data Loss Prevention [Электронный ресурс]. URL: https://www.broadcom.com/products/cyber-security/information-protection/data-loss-prevention (дата обращения: 11.10.2024).
57. Tan L., Hu K., Zhou X., Chen R., Jiang W. Print-scan invariant text image watermarking for hardcopy document authentication // Multimedia Tools and Applications. — 2019. — Т. 78. — С. 13189-13211.
58. Tan M., Le Q. EfficientNet: rethinking model scaling for convolutional neural networks // International conference on machine learning. — PMLR, 2019. — С. 6105–6114.
59. Topkara M., Topkara U., Atallah M. J. Words are not enough: Sentence level natural language watermarking // In: Proceedings of the 4th ACM International Workshop on Contents Protection and Security, Xi'an, China, 30 May 2006.
60. TRACE DOC. Система расследования утечек информации [Электронный ресурс]. URL: https://secretgroup.ru/nashi-produkty/trace-doc/ (дата обращения: 11.10.2024).
61. Wang T.H., Huang H. J., Lin J.-T. Omnidirectional CNN for visual place recognition and navigation // 2018 IEEE International Conference on Robotics and Automation (ICRA). — IEEE, 2018. — С. 2341–2348.
62. Window.Topmost. Свойство [Электронный ресурс]. URL: https://learn.microsoft.com/ru-ru/dotnet/api/system.windows.window.topmost?view=windowsdesktop-8.0 (дата обращения: 11.10.2024).
63. Xiao C., Zhang C., Zheng C. Fontcode: Embedding information in text documents using glyph perturbation // ACM Transactions on Graphics (TOG). — 2018. — Т. 37, No 2. — С. 1—16.
64. XPSDrv Render Module [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/windows-hardware/drivers/print/xpsdrv-render-module (дата обращения: 11.10.2024).
65. Yakushev A., Markin Y., Obydenkov D., Frolov A., Fomin S., Akopyan M., Kozachok A., Gaynov A. Docmarking: Real-Time Screen-Cam Robust Document Image Watermarking // 2022 Ivannikov Ispras Open Conference (IVMEM). — 2022. — P. 142–150.

66. Zharikov, I., Nikitin, P., Vasiliev, I., & Dokholyan, V. DDI-100: dataset for text detection and recognition. In Proceedings of the 2020 4th International Symposium on Computer Science and Intelligent Control. — 2020. — C. 1–5.

67. Zhou W. Image quality assessment: from error measurement to structural similarity // IEEE Transactions on Image Processing. — 2004. — Vol. 13. — P. 600–613.

Приложение А

Библиотека маркирования текстовых документов на экране путем изменения яркости в областях межстрочных интервалов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2020667628

«Библиотека маркирования текстовых документов на экране путем изменения яркости в областях межстрочных интервалов»

Правообладатель: ***Федеральное государственное бюджетное учреждение науки Институт системного программирования им. В.П. Иванникова Российской академии наук (RU)***

Авторы: ***см. на обороте***

Заявка № **2020666863**

Дата поступления **17 декабря 2020 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***25 декабря 2020 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

Модуль маркирования текстовых документов при печати для ОС семейства Linux

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2020667580

«Модуль маркирования текстовых документов при печати для ОС семейства Linux»

Правообладатель: ***Федеральное государственное бюджетное учреждение науки Институт системного программирования им. В.П. Иванникова Российской академии наук (RU)***

Авторы: ***Маркин Юрий Витальевич (RU), Козачок Александр Васильевич (RU), Фомин Станислав Александрович (RU), Акопян Манук Сосович (RU), Обыденков Дмитрий Олегович (RU), Горбачев Павел Николаевич (RU), Козлов Сергей Викторович (RU), Громей Дмитрий Дмитриевич (RU), Копылов Сергей Александрович (RU), Падарян Вартан Андроникович (RU)***

Заявка № **2020666720**

Дата поступления **17 декабря 2020 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***24 декабря 2020 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

Библиотека маркирования текстовых документов при печати за счет горизонтального смещения слов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2020667592

«Библиотека маркирования текстовых документов при печати за счет горизонтального смещения слов»

Правообладатель: ***Федеральное государственное бюджетное учреждение науки Институт системного программирования им. В.П. Иванникова Российской академии наук (RU)***

Авторы: ***см. на обороте***

Заявка № **2020666902**
Дата поступления **17 декабря 2020 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***24 декабря 2020 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

Модуль маркирования текстовых документов при печати для ОС семейства Windows

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2020667579

«Модуль маркирования текстовых документов при печати для ОС семейства Windows»

Правообладатель: ***Федеральное государственное бюджетное учреждение науки Институт системного программирования им. В.П. Иванникова Российской академии наук (RU)***

Авторы: ***Маркин Юрий Витальевич (RU), Козачок Александр Васильевич (RU), Фомин Станислав Александрович (RU), Акопян Манук Сосович (RU), Обыденков Дмитрий Олегович (RU), Горбачев Павел Николаевич (RU), Козлов Сергей Викторович (RU), Громей Дмитрий Дмитриевич (RU), Копылов Сергей Александрович (RU), Кондратьев Борис Владимирович (RU), Падарян Вартан Андроникович (RU)***

Заявка № **2020666721**

Дата поступления **17 декабря 2020 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***24 декабря 2020 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

Модуль маркирования текстовых документов на экране для ОС семейства Windows

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2020667308

«Модуль маркирования текстовых документов на экране для ОС семейства Windows»

Правообладатель: ***Федеральное государственное бюджетное учреждение науки Институт системного программирования им. В.П. Иванникова Российской академии наук (RU)***

Авторы: ***Маркин Юрий Витальевич (RU), Козачок Александр Васильевич (RU), Фомин Станислав Александрович (RU), Акопян Манук Сосович (RU), Обыденков Дмитрий Олегович (RU), Якушев Алексей Юрьевич (RU), Падарян Вартан Андроникович (RU)***

Заявка № **2020666722**

Дата поступления **17 декабря 2020 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***22 декабря 2020 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

Приложение Б

Акты о внедрении результатов диссертации

исх. № 612/0924 от 29.09.2024г.

Общество с ограниченной ответственностью
«Системы и Технологии»
Юридический адрес: 123290, г. Москва, 1-й Магистральный тупик, д. 5А
Почтовый адрес: 123290, г. Москва, 1-й Магистральный тупик, д. 5А
ОКПО: 17376123, ОГРН: 1157746182550, ИНН: 7713392106,
КПП: 771401001, e-mail: info@systech.msk.ru

АКТ

о внедрении результатов кандидатской диссертационной работы

Обыденкова Дмитрия Олеговича

Результаты диссертационного исследования Обыденкова Дмитрия Олеговича на тему «Методы противодействия анонимности при утечках текстовых документов посредством цифровых водяных знаков» внедрены в цифровом контуре Общества с ограниченной ответственностью «Системы и Технологии» в виде модуля обнаружения утечки данных ПС «МОУД» ЦРПМ.30005-01. МОУД позволяет деанонимизировать утечки изображений конфиденциальных текстовых документов, реализованные посредством фотографирования документов на экране монитора, а также фотографирования или сканирования распечатанных документов.

Генеральный директор
ООО «СиТ» А.Ф. Попов

«29» сентября 2024 года