Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа ядерных технологий
Отделение экспериментальной физики
Направление подготовки: Прикладная математика и информатика

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Нейросетевое распознавание данных финансовой отчетности компаний России

УДК 004.7.032.26:657.6

Студент

Группа	ФИО	Подпись	Дата
0ВМ92	Кулигин Сергей Михайлович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Крицкий О. Л.	к.ф.-м.н.		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Киселева Е.С.	к.э.н		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Антоневич О.А.	к.б.н		

ДОПУСТИТЬ К ЗАЩИТЕ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Семёнов М. Е.	к.ф.-м.н.		

Томск – 2021 г

Планируемые результаты обучения по ООП

Код результата	Результат обучения
ПК(У)-1	Способен проводить научные исследования и получать новые научные и прикладные результаты самостоятельно и в составе научного коллектива
ПК(У)-2	Способен проводить поиск и анализ научной и научно-технической литературы по тематике проводимых исследований
ПК(У)-3	Способен разрабатывать и анализировать показатели качества информационных систем, используемых в производственной деятельности
ПК(У)-4	Способен планировать научно-исследовательскую деятельность, анализировать риски, управлять проектами, управлять командой проекта
ПК(У)-5	Способен к преподаванию математических дисциплин и информатики в образовательных организациях высшего образования
ПК(У)-6	Способен проектировать и организовывать учебный процесс по образовательным программам с использованием современных образовательных технологий
ОПК(У)-1	Способен решать актуальные задачи фундаментальной и прикладной математики
ОПК(У)-2	Способен совершенствовать и реализовывать новые математические методы решения прикладных задач
ОПК(У)-3	Способен разрабатывать математические модели и проводить их анализ при решении задач в области профессиональной деятельности
ОПК(У)-4	Способен комбинировать и адаптировать существующие информационно-коммуникационные технологии для решения задач в области профессиональной деятельности с учетом требований информационной безопасности
УК(У)-1	Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий
УК(У)-2	Способен управлять проектом на всех этапах его жизненного цикла
УК(У)-3	Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели
УК(У)-4	Способен применять современные коммуникативные технологии, в том числе на иностранном(-ых) языке(-ах), для академического и профессионального взаимодействия
УК(У)-5	Способен анализировать и учитывать разнообразие культур в процессе межкультурного взаимодействия
УК(У)-6	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
0ВМ92	Кулигину Сергею Михайловичу

Школа	ИЯТШ	Отделение (НОЦ)	ОЭФ
Уровень образования	Магистратура	Направление/специальность	Прикладная математика и информатика

Тема ВКР:

«Нейросетевое программирование финансовой отчетности компаний России»	
Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Применяется методология классификации изображений в соответствии с разработанной системой классификационных признаков машинного распознавания таблиц для выявления финансовой информации. Используется технология машинного обучения нейронной сети. Рабочая зона – помещение с персональными компьютерами.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	– ГОСТ Р 50923-96 Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. – ГОСТ 12.2.032-78 Система стандартов безопасности труда. – ГОСТ Р ИСО 9241-2-2009 Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов.
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	Рассмотрим вредные факторы: – отклонение показателей микроклимата рабочей зоны; – недостаточная освещенность рабочей зоны; – отсутствие или недостаток естественного света; – повышенный уровень шума на рабочем месте; – Повышенное образование электростатических зарядов
3. Экологическая безопасность:	– анализ воздействия при работе на ПЭВМ на атмосферу, гидросферу, литосферу; – наличие отходов (бумага,

	картриджи, компьютеры и т. д.); методы утилизации отходов.
4. Безопасность в чрезвычайных ситуациях:	– возможные ЧС – перебои в электроснабжении и связи; – типичная ЧС – пожар на рабочем месте.

Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Антоневич О.А.	к.б.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0ВМ92	Кулигин Сергей Михайлович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
0ВМ92	Кулигину Сергею Михайловичу

Школа	ИЯТШ	Отделение школы (НОЦ)	ОЭФ
Уровень образования	Магистратура	Направление/специальность	01.04.02 «Прикладная математика и информатика»

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. *Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих*	*Стоимость материальных ресурсов определялась в соответствии с рыночными ценами г. Томска. Тарифные ставки исполнителей определены штатным расписанием НИ ТПУ.*
2. *Нормы и нормативы расходования ресурсов*	*Коэффициенты для расчета заработной платы.*
3. *Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования*	*Коэффициент отчислений во внебюджетные фонды – 30,2 %*
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. *Оценка коммерческого и инновационного потенциала НТИ*	*1.Потенциальные потребители результатов исследования;* *2.SWOT – анализ;* *3.Оценка готовности проекта к коммерциализации*
2. *Разработка устава научно-технического проекта*	*1.Постановка цели, ожидаемых результатов проекта;* *2.Определение внутренних и внешних заинтересованных сторон проекта;* *3.Определение ограничений/допущений проекта.*
3. *Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок*	*1.Определение структуры и трудоемкости выполнения работ;* *2.Бюджет научно - технического исследования (НТИ);* *3. Реестр рисков проекта*
Перечень графического материала *(с точным указанием обязательных чертежей):*	
1. Сегментирование рынка *2. Матрица SWOT* *3. Оценка готовности проекта к коммерциализации* *4. Заинтересованные стороны* *5. Цели и результат проекта и рабочая группа проект* *6. Ограничения/допущения проекта* *7. Иерархическая структура работ проекта* *8. Комплекс работ по разработке проекта* *9. Временные показатели осуществления комплекса работ* *10. Календарный план-график выполнения работ (диаграмма Гантта)* *11. Расчёт бюджета исследования* *12. Реестр рисков*	

Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Киселева Е.С.	к.э.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0ВМ92	Кулигин Сергей Михайлович		

Реферат

Пояснительная записка к магистерской диссертации выполнена на 103 страницах машинописного текста, содержит 23 таблицы, 17 рисунков, 38 формул, 24 источников, 2 приложения.

Ключевые слова: распознавание, OpenCV, Tesseract, рекуррентная нейронная сеть.

Объект исследования: изображения с таблицами, содержащих какую-то информацию.

Цель исследования: распознавание таблиц с информацией с помощью нейронных сетей и запись её в файл.

Методы проведения исследования: теоретические и практические.

Полученные результаты: с помощью компьютерного зрения и модели нейронной сети получен файл с необходимой информацией.

Оглавление

Введение

Актуальность. Не смотря на то, что большинство профессий, о которых мы задумываемся, не связаны с компьютерными программами и вычислениями, в современном обществе практически каждая такая профессия завязана хоть даже на самом минимальном использовании какого-либо программного обеспечения. Так, в области экономики, при осуществлении расчётов используются реализации алгоритмов этих расчётов с помощью программного кода. Соответственно, вся информация о значениях переменных алгоритма должна быть представлена в цифровом виде. Обычно эта информация содержится в финансовых отчётах, которые сканируются и присылаются специалистам, осуществляющих расчёт. Но на этом этапе возникает проблема, которая состоит в том, что отсканированный финансовый отчёт представляет собой совокупность изображений без возможности работы с текстом и, как следствие, без возможности быстро перенести информацию для расчёта в программу, поэтому все данные приходится вносить вручную. Для решения проблемы необходимо каким-то образом распознать информацию, в данном случае таблицы с данными, чтобы существенно упростить работу с ней.

Механизмы распознавания нового поколения действительно хорошо справляются с этими проблемами, используя новейшие исследования в области глубокого обучения. Используя комбинацию глубоких моделей и общедоступных огромных наборов данных, модели достигают высочайшей точности при выполнении поставленных задач.

Целью магистерской диссертации является реализация алгоритма по распознаванию таблиц и текста, внесённого в них, на изображении.

Для достижения поставленной цели необходимо решить следующие **задачи**:

1. Реализовать распознавание таблиц на изображении;
2. Обучить нейронную сеть для распознавания текста;
3. Провести тестирование полученной реализации.

Основные термины

Рекуррентные нейронные сети (англ. Recurrent neural network; RNN) – вид нейронных сетей, где связи между элементами образуют направленную последовательность.

Нейрон – это единица сети, которая на входе получает сигнал на основании заданных параметров производит с ними вычислительные действия и передает их либо на следующий слой, либо на выход.

Финансовая информация – это информация, раскрывающая экономическое состояние рассматриваемого объекта, т.е. его некое описание на языке чисел.

1. Теоретическая часть

1.1 OpenCV

OpenCV (Open Source Computer Vision) – самая важная библиотека с открытым исходным кодом в области компьютерного зрения. Она не только включает множество алгоритмов анализа и обработки изображений, но также включает классические алгоритмы машинного обучения и библиотеки алгоритмов глубокого обучения. Эти алгоритмы машинного обучения играют ключевую роль в задачах компьютерного зрения, таких как классификация изображений, обнаружение целей, отслеживание целей, а также оптическое обнаружение и распознавание символов [1].

При обработке изображений входом является изображение, а выходом обычно является другое изображение. Выходное изображение иногда является изменённой версией входного изображения. Иногда это обработанная версия входного изображения, так что на выходе получается упрощенная версия входа. В других случаях это сжатая версия изображения.

В компьютерном зрении входом является изображение, а выходом – информация, содержащаяся в нём. Например, в стереофоническом алгоритме входом является пара изображений, а выходом – карта глубины. Пример такой карты представлен на рисунке 1.1.

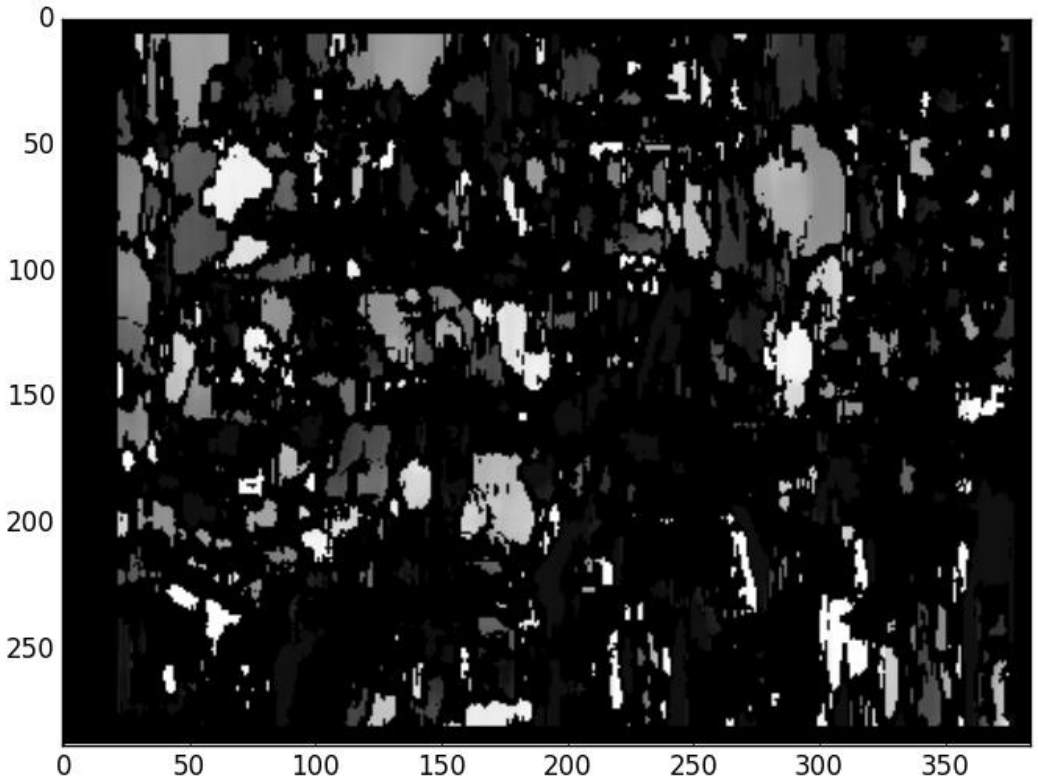

Рисунок 1.1 – Пример карты глубины.

Результатом также может быть метка класса, как мы видим в классификации изображений, или набор ограничивающих рамок и меток классов, как мы видим при обнаружении объектов.

В рамках магистерской диссертации будут использоваться бинаризация и морфологические преобразования для обработки изображения. Бинаризация по своей сути – это перевод изображения из цветного в монохромное: только с черными и белыми пикселями. Есть различные методы бинаризации, которые условно можно разделить на адаптивные и пороговые. Адаптивные методы не принимают фиксированное значение порога для всего изображения, а пороговое значение вычисляется для меньших областей, так что для разных областей изображения будут разные значения. Пороговые методы бинаризации определяют фиксированное значение порога. Это означает, что мы оставим только те пиксели, интенсивность которых выше порогового значения. Если передаваемое

значение интенсивности при установлении порога заменяется максимально возможным значением интенсивности (белый), а другие заменяются самым низким значением (чёрный), то есть нулевым, мы получаем изображение только в двух цветах. Рассмотрим бинаризацию на примере изображения, представленного на рисунке 1.2.

Рисунок 1.2 – Пример исходного изображения.

Сначала изображение нужно перевести в градации серого с помощью функции $cvCvtColor$, а затем выбрать порог яркости. Например, на рисунке 1.3 выбран порог яркости 128.

Рисунок 1.3 – Пример бинаризации с порогом яркости 128.

Каждый раз вручную для каждого изображения подбирать свой порог яркости неудобно, поэтому существуют различные методы для расчёта оптимального порога бинаризации, например, Отсу, Бернсена, Эйквеля, Ниблэка и т.п. Самым быстрым и качественным считается метод Отсу.

Метод Отсу используется для выполнения автоматической пороговой обработки изображения. Это эквивалентно глобально оптимальному алгоритму k-средних, выполняемому на гистограмме интенсивности. Этот метод использует как простой, так и адаптивный порог, имея единое пороговое значение для всего изображения и удаляя нежелательный фон, который может возникнуть в адаптивном пороге. Порог определяется путём минимизации дисперсии интенсивности внутри класса или, что эквивалентно, путём максимизации дисперсии между классами, которая выражается в терминах вероятности ω_iи среднего арифметического класса μ_i, которое, в свою очередь, может обновляться итеративно [1]:

$$\sigma_\omega^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t), \quad (1)$$

$$\sigma_b^2(t) = \sigma^2 - \sigma_\omega^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2, \quad (2)$$

Эта идея привела к эффективному алгоритму:

1. Найти гистограмму h изображения и вычислить частоту V для каждого уровня яркости монохромного изображения G;
2. Вычислить начальные значения $\omega_1(0), \omega_2(0)$ и $\mu_1(0), \mu_2(0)$;
3. Для каждого значения t полутона:
   3.1. Обновляем $\omega_1(t), \omega_2(t)$ и $\mu_1(t), \mu_2(t)$;
   3.2. Вычисляем $\sigma_b^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2$;
   3.3. Если $\sigma_b^2(t)$ больше, чем уже имеющееся, то запоминаем $\sigma_b^2(t)$ и значение порога t.
4. Искомый порог соответствует максимуму $\sigma_b^2(t)$.

Используя этот алгоритм, мы можем найти пороговое значение яркости для изображения на рисунке 1.2. Это значение будет равно 151.

Рисунок 1.4 – Пример бинаризации с использованием критерия Отсу.

Как видно из рисунка 1.4, на исходном изображении слишком много светлых мест, поэтому порог сдвинулся в светлую сторону, и в результате некоторые окна почти полностью стали чёрного цвета.

Среди адаптивных подходов к бинаризации следует отметить подходы, реализованные в функции *cvAdaptiveThreshold*. Эта функция преобразует изображение в градациях серого к монохромному изображению согласно формулам:

$$\text{CV\_THRESH\_BINARY} \quad dst(x,y) = \begin{cases} \max if\ \text{src}(x,y) > T(x,y) \\ 0\ otherwise \end{cases}, \tag{3}$$

$$\text{CV\_THRESH\_BINARY\_INV} \quad dst(x,y) = \begin{cases} 0\ if\ \text{src}(x,y) > T(x,y) \\ \max otherwise \end{cases}, \tag{4}$$

где $T(x,y)$ – порог, рассчитываемый индивидуально для каждого пикселя.

1.2 Tesseract OCR

OCR (Optical Character Recognition) – это оптическое распознавание символов. Другими словами, системы OCR преобразуют двумерное изображение текста, которое может содержать машинно-напечатанный или рукописный текст, из его графического представления в машиночитаемый текст. OCR как процесс обычно делится на несколько подпроцессов, которые должны выполняться с максимальной точностью [3]:

- Предварительная обработка изображения;
- Текстовая локализация;
- Сегментация;
- Распознавание;
- Постобработка.

Подпроцессы в приведенном выше списке могут отличаться, но это примерные этапы, необходимые для достижения автоматического распознавания символов. В OCR его основная цель – идентифицировать и фиксировать все уникальные слова с использованием разных языков из письменных текстовых символов. В рамках магистерской диссертации будет

использоваться оболочка для Tesseract OCR – Pytesseract. Tesseract включает подсистему нейронной сети, настроенную как распознаватель текстовых строк. Она была разработана на основе модели OCRopus на Python, которая была ответвлением LSMT (Long short-term memory – долгая кратко-срочная память) на C++ под названием CLSTM. CLSTM – это реализация модели рекуррентной нейронной сети LSTM на C++ с использованием библиотеки Eigen для численных вычислений. Процесс распознавания текста в Tesseract представлен на рисунке 1.5.

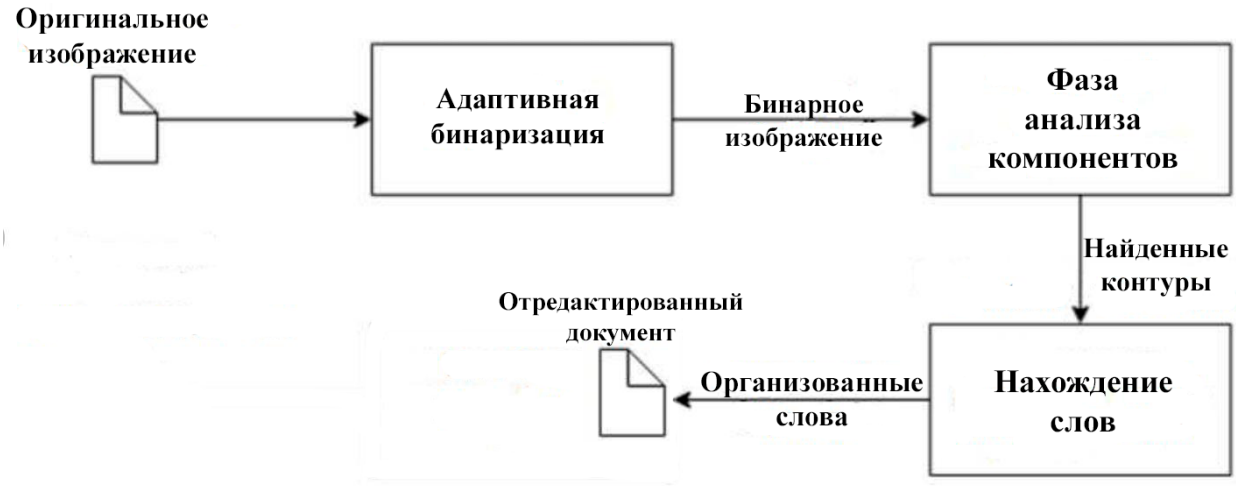

Рисунок 1.5 – Процесс распознавания текста в Tesseract.

Tesseract зависит от многоступенчатого процесса, в котором можно различать шаги:

- Поиск слов;
- Поиск линий;
- Классификация символов.

Поиск слов осуществляется путём заключения текстовых строк в небольшие прямоугольники, а строки и прямоугольники анализируются на предмет фиксированного шага или пропорционального текста. Текстовые строки разбиваются на слова в зависимости от интервала между символами. Затем распознавание осуществляется в два шага. Сначала делается попытка распознать каждое слово по очереди. Одобренные слова передаются в

адаптивный классификатор в качестве обучающих данных. Затем адаптивный классификатор получает возможность лучше распознавать текст.

Чтобы избежать всех возможных падений точности вывода, необходимо убедиться, что изображение предварительно обработано надлежащим образом. Это включает изменение масштаба, бинаризацию, удаление шума, выравнивание и т.д.

1.3 Нейронные сети

Нейронные сети и в настоящее время предоставляют лучшие решения многих проблем в области распознавания изображений, речи и обработки естественного языка.

Нейронная сеть основана на совокупности связанных элементов, называемых нейронами, которые в общих чертах являются моделями нейронов в биологическом мозге. Нейроны могут передавать сигналы друг другу с помощью своих соединений, как синапсы в биологическом мозге. Нейрон, который получает сигнал, обрабатывает его и может сигнализировать подключенным к нему нейронам. Сигнал в соединении – это некое действительное число, и выходной сигнал каждого нейрона вычисляется некоторой нелинейной функцией суммы его входов [4].

Нейроны и связи обычно имеют вес, который корректируется по мере обучения. Вес напрямую влияет на степень важности сигнала в соединении. Нейроны могут иметь такой порог, что сигнал отправляется только в том случае, если совокупный сигнал пересекает этот порог.

Обычно нейроны объединены в слои, на каждом из которых могут выполняться разные преобразования на входе. Пример небольшой простой традиционной нейронной сети представлен на рисунке 1.6.

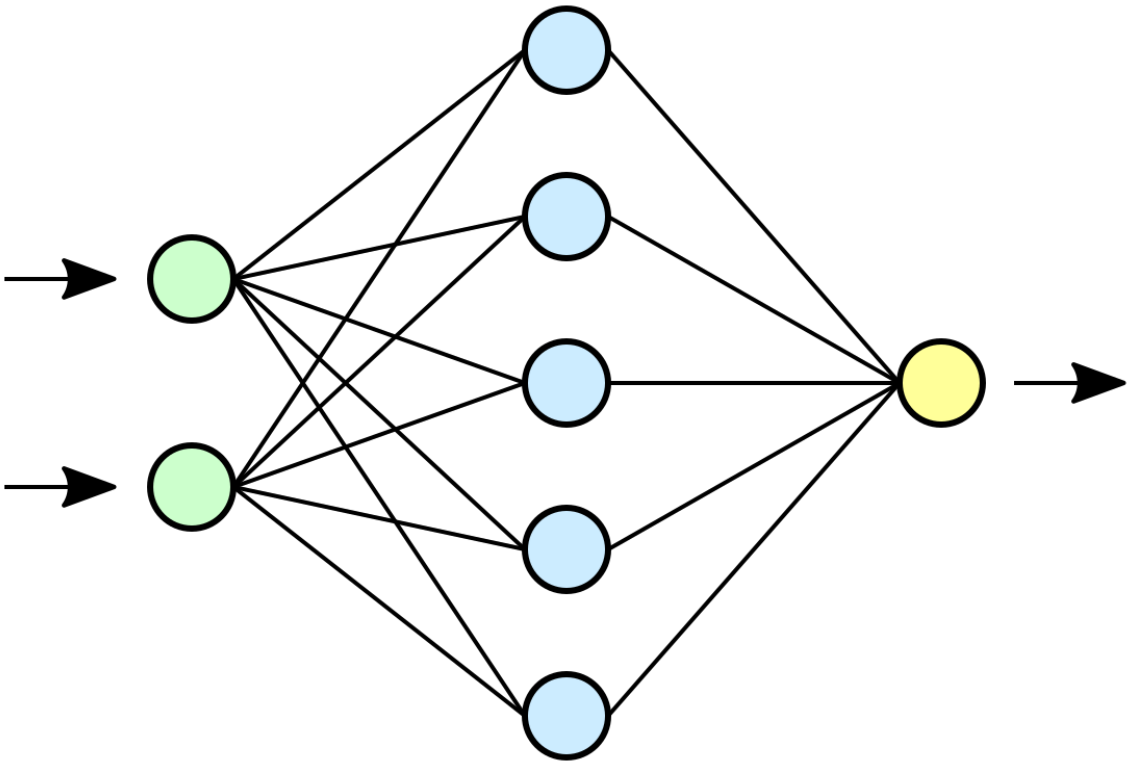

Рисунок 1.6 – Схематичное изображение традиционной нейронной сети

Основными направлениями применения нейронных сетей является решение следующих задач:

- Классификация – распределение данных по параметрам;
- Предсказание – возможность предсказывать следующий шаг;
- Распознавание – возможность на основании расположения пикселей распознавать то или иное изображение на рисунке или фото.

В традиционных нейронных сетях есть один большой недостаток. Они не могут использовать свои же результаты на предыдущих итерациях, чтобы использовать их на более поздних, так как синапсы в них последовательны. Для решения этой проблемы существуют рекуррентные нейронные сети. Внутри таких сетей информация может сохраняться, так как она разбита на итерации и на каждой новой итерации входом являются как новые считываемые данные, так и выход предыдущей итерации. Таким образом, сеть может проверять считываемую информацию, сопоставляя с уже

распознанной. Именно такая сеть будет использоваться в рамках научной работы, пример приведён на рисунке 1.7.

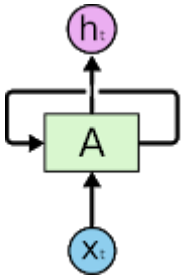

Рисунок 1.7 – Схематичное изображение рекуррентной нейронной сети.

На рисунке 1.7 на вход подаётся фрагмент информации x_t, проходит обработку в элементе нейронной сети A и выводи значение h_t. Также имеется цикл с помощью которого информация передаётся от одной итерации к следующей.

Если развернуть рекуррентную нейронную сеть, то можно увидеть, что она состоит из нескольких копий одной и той же сети, каждая из которых передает информацию другой.

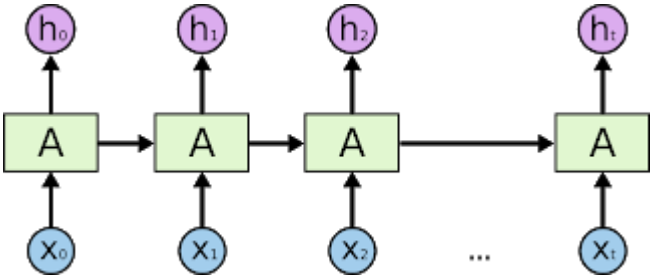

Рисунок 1.8 – Развёрнутая рекуррентная нейронная сеть.

К сожалению, у рекуррентных сетей есть одна серьёзная проблема – неспособность обрабатывать долгосрочные зависимости. Одним из преимуществ рекуррентных сетей является идея о том, что они могут соединить предыдущую информацию с текущей задачей, но если разрыв между ними будет слишком большим, сеть не сможет научиться связывать информацию.

Сети с долговременной краткосрочной памятью (LSTM) – это особый вид рекуррентных нейронных сетей, способный изучать долгосрочные зависимости [5]. LSTM специально разработаны, чтобы избежать проблемы

долгосрочной зависимости. Они также имеют структуру, подобную цепочке, но повторяющийся модуль имеет другую структуру. Вместо одного слоя нейронной сети существует четыре, взаимодействующих особым образом.

1.4 Алгоритм Рамера-Дугласа-Пекера

Алгоритм Рамера-Дугласа-Пекера применим в основном для решения задач упрощения полигональной цепи. Такой вид задач широко распространен при построении карт и при обработке векторной графики. В качестве примера можно взять ломаную линию, несколько точек соединения которой попадают в одно и то же место – очевидно, что все эти точки можно убрать, оставив только одну.

Цель алгоритма – создать упрощенную полилинию, которая имеет меньше точек, чем исходная, но при этом сохраняет форму оригинала.

На вход алгоритма подаются следующие данные: координаты всех точек между первой и последней включительно; максимальное расстояние ε, которое может быть между исходной и упрощенной полилиниями. Затем алгоритм рекурсивно делит полилинию, при этом первая и последняя точка сохраняются. Теорема о расстоянии от точки до прямой на плоскости применяется для нахождения точки, которая наиболее удалена от отрезка, состоящего из первой и последней точек.

Теорема. Если прямая проходит через две точки (x_1, y_1) и (x_2, y_2), то расстояние от точки (x_0, y_0) до прямой равно:

$$\frac{|(y_2-y_1)x_0-(x_2-x_1)y_0+x_2y_1-y_2x_1|}{\sqrt{(y_2-y_1)^2+(x_2-x_1)^2}}. \quad (5)$$

Знаменатель этого выражения равен расстоянию между точками (x_1, y_1) и (x_2, y_2). Числитель равен удвоенной площади треугольника с вершинами (x_0, y_0), (x_1, y_1) и (x_2, y_2). Выражение эквивалентно $h = \frac{2A}{b}$, что может быть получено преобразованием стандартной формулы площади

треугольника: $A = \frac{1}{2}bh$, где b — длина стороны, а h — высота на эту сторону из противолежащей вершины.

Если точка находится на расстоянии, меньшем, чем ε, то все точки, которые еще не были отмечены к сохранению, могут быть выброшены из набора, и получившаяся прямая сглаживает кривую с точностью не ниже ε. Если же это расстояние больше ε, то алгоритм рекурсивно вызывает себя на наборе от начальной точки до данной и от данной до конечных точек. Стоит отметить, что алгоритм Рамера-Дугласа-Пекера в ходе своей работы не сохраняет топологию. Это означает, что в результате мы можем получить линию с самопересечениями [6].

В качестве наглядного примера возьмем полилинию со следующим набором точек: $[\{1; 5\},\{2; 3\},\{5; 1\},\{6; 4\},\{9; 6\},\{11; 4\},\{13; 3\},\{14; 2\},\{18; 5\}]$ и посмотрим на процесс упрощения при разных значениях ε, представленный на рисунках 1.9-1.12:

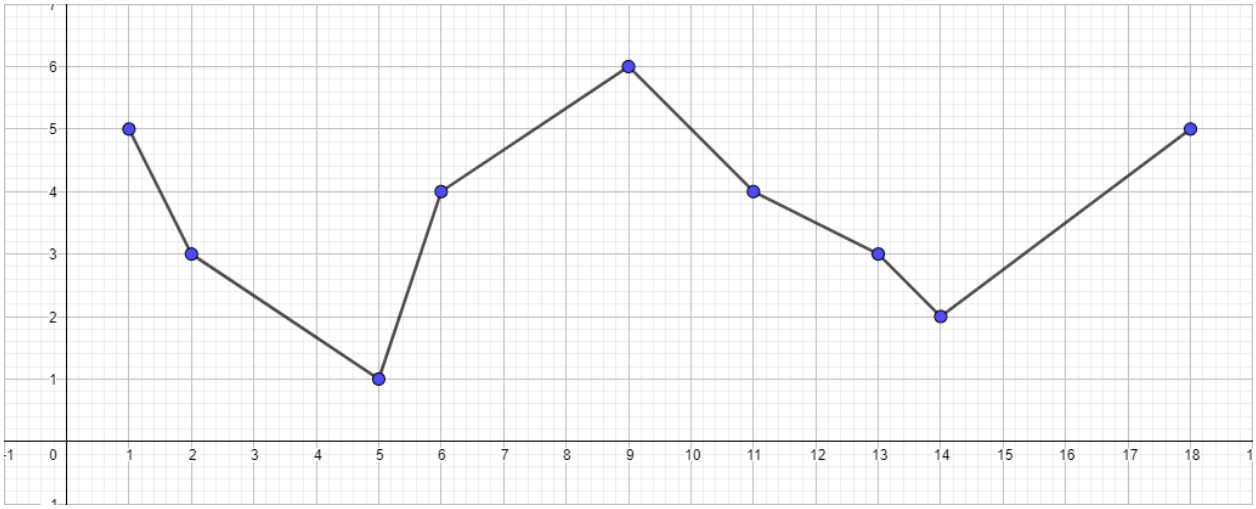

Рисунок 1.9 – Исходная полилиния из представленного набора точек.

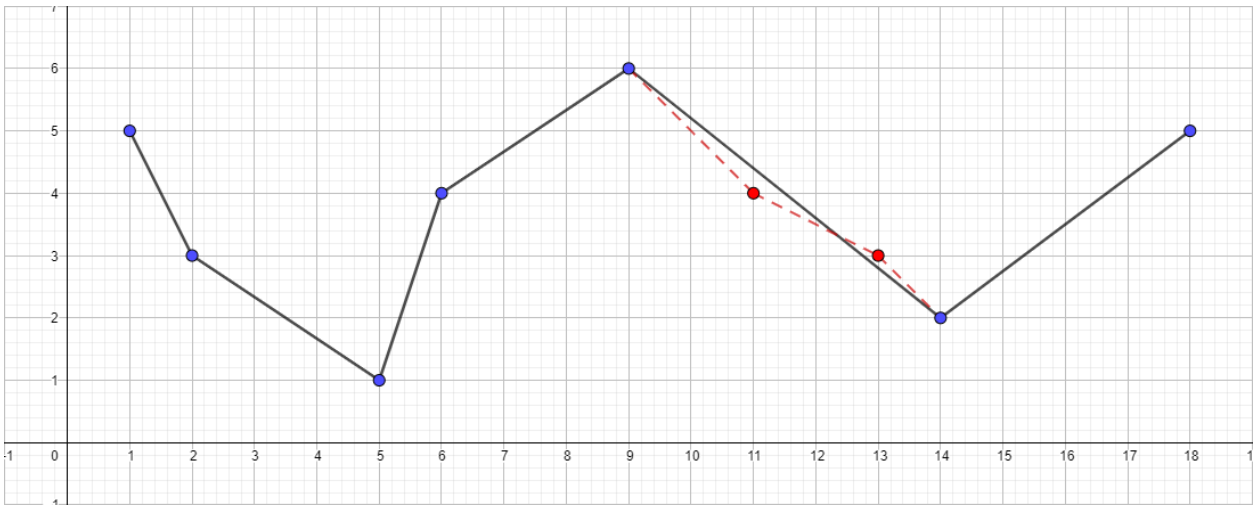

Рисунок 1.10 – Полилиния с ε равной 0.5.

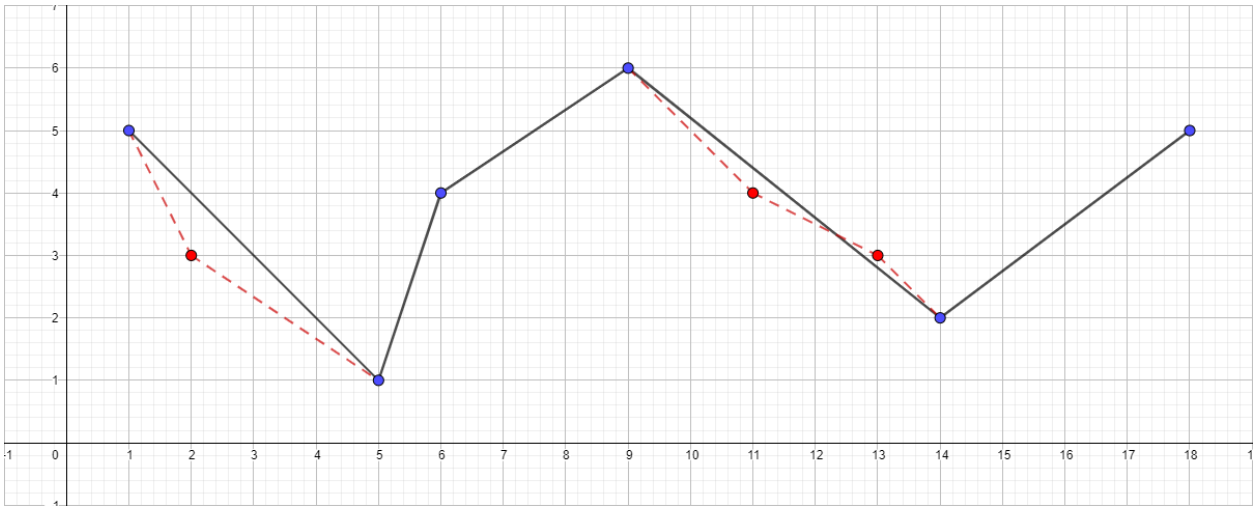

Рисунок 1.11 – Полилиния с ε равной 1.

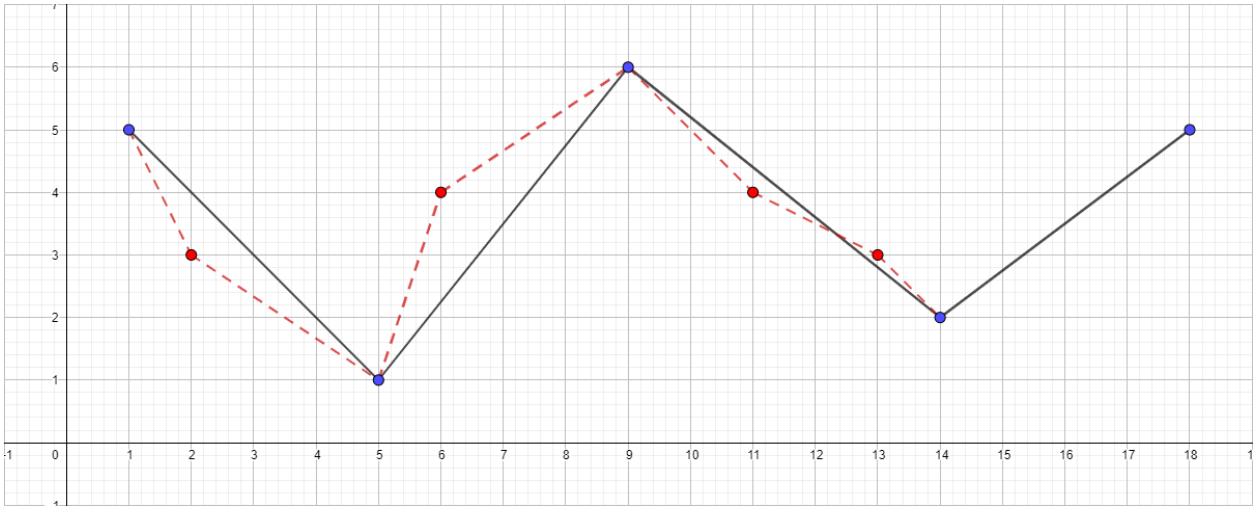

Рисунок 1.12 – Полилиния с ε равной 1.5.

Говоря о сложности алгоритма, он работает за $O(n^2)$, но можно внести дополнения, которые позволят получить $O(nlogn)$. Ускорение основывается

на уменьшении времени поиска наиболее удаленной вершины. Это можно осуществить благодаря идее о том, что наиболее удаленная вершина лежит на выпуклой оболочке полигональной цепи. Для построения выпуклой оболочки используется алгоритм Мелкмана, который работает для двумерной полигональной цепи без самопересечений за $O(n)$, строя все промежуточные выпуклые оболочки, которые пригодятся в дальнейшем. После построения выпуклой оболочки используется бинарный поиск для нахождения наиболее удаленной вершины за $O(logn)$. Затем, если потребуется, действия повторятся рекурсивно, как и в оригинальном алгоритме, но заново строить оболочки не имеет смысла, так как промежуточные уже были построены. Использовав их, можно разбить текущую оболочку на две за $O(logn)$. В итоге получается, $O(n)$ разбиений за $O(logn)$ и поиск за $O(logn)$, что даёт $O(nlogn)$.

К сожалению, у данного метода есть несколько недостатков по сравнению с оригинальным алгоритмом, некоторые из которых уже были упомянуты:

- Исходная цепь должна быть без самопересечений для использования алгоритма Мелкмана;
- Ускорение подходит лишь для двумерного случая из-за способа построения выпуклой оболочки;
- В некоторых случаях оригинальный алгоритм Рамера-Дугласа-Пекера работает быстрее, например, в случае, когда цепь приближено является окружностью.

2. Практическая часть

2.1 Выбор среды программирования

Одним из важных аспектов качетвенной реализации модели явлется выбор оптимальной программной среды. В настоящее время существует большое количество языков программирования и сред разработки.

В рамках магистерской диссертации необходимо запрограммировать распознавание таблиц с изображения и запись информации в файл. Программирование будет осуществляться с помощью языка Python. Python – это язык программирования общего назначения, который очень быстро стал очень популярным, в основном из-за его простоты и читабельности кода.

По сравнению с такими языками, как C/C++, Python медленнее. Тем не менее, Python можно легко расширить с помощью C/C++, что позволяет нам писать ресурсоёмкий код на C/C++ и создавать оболочки Python, которые можно использовать как модули Python [7]. Это дает два преимущества: во-первых, код работает так же быстро, как и исходный код C/C++, поскольку это фактический код C++, работающий в фоновом режиме, и, во-вторых, его легче кодировать на Python, чем на C/C++.

OpenCV использует библиотеку Numpy, которая является высоко оптимизированной библиотекой для числовых операций с синтаксисом в стиле MATLAB. Все структуры массивов OpenCV преобразуются в массивы Numpy и обратно. Это также упрощает интеграцию с другими библиотеками, использующими Numpy, такими как SciPy и Matplotlib.

2.2 Реализация распознавания

Для решения поставленной задачи в основном будет использоваться библиотека для обработки изображений OpenCV, а также библиотека для распознавания текста Tesseract OCR. В свою очередь, Tesseract OCR использует нейронные сети для поиска и распознавания текста на

изображении и включает в себя обученные языковые модели и разные виды распознавания [9].

Реализация распознавания необходимых объктов, в данном случае таблиц, и запись информации из них в файл состоит из нескольких этапов, таких как:

- Загрузка изображения;
- Фильтрация изображения;
- Изоляция линий;
- Выделение таблиц;
- Запись в файл.

Перед запуском программы ей будет передаваться путь к файлу, который необходимо обработать.

2.2.1 Загрузка и фильтрация изображения

Говоря о загрузке файла ограничимся форматами файлов *pdf* и *jpg,*так как эти форматы являются основными при работе с программой. Если формат файла не соответствует ни одному из перечисленных, то будет выведено сообщение «Должен использоваться формат pdf или jpg.» и осуществляться выход из программы. Чтение файла осуществляется с помощью метода *cv.imread*. Файл, который будем загружать, представлен на рисунке 2.1.

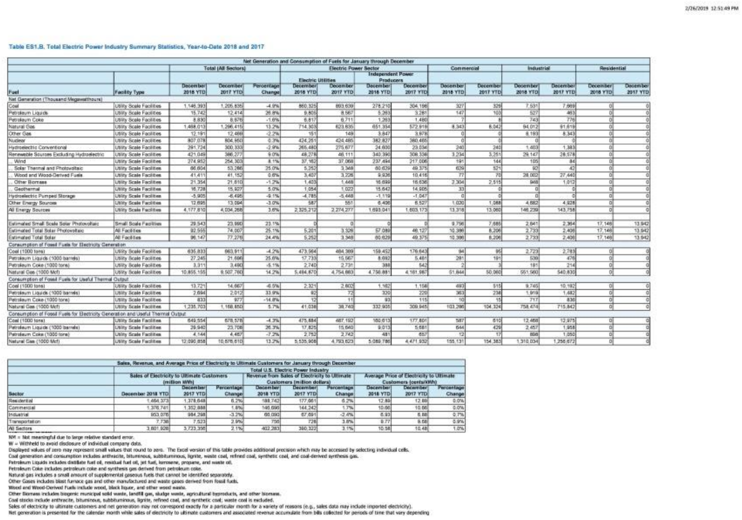

Рисунок 2.1 – Входные данные.

После загрузки файла будет создана его копия в формате *png* и именем «*target*» и далее для работы будет использоваться именно он. Это сделано для того, чтобы исключить возможность потери исходного файла путём его редактирования в ходе выполнения программы.

Загруженное изображение в программе будет представленно в виде матрицы размером mxn

$$\begin{pmatrix} 1 & \dots & n \\ \dots & \dots & \dots \\ m & \dots & mn \end{pmatrix}, \tag{6}$$

где m – количество пикселей по вертикали; n – количество пикселей по горизонтали. Каждый отдельный элемент матрицы представлен в виде числа от 0 до 255, которое является значением цветности пикселя.

В основном, для всех изображений используется три канала для передачи цвета: красный, зелёный и синий. В итоге, в программе каждое изображение будет закодировано тремя такими матрицами (для каждого из каналов цветности), поэтому для успешной реализации обработки изображения необходимо перевести исходное изображение в градации серого. После этого останется только один параметр цветности – яркость, который может принимать значения от 0 до 255.

Фильтрация изображения подразумевает, что в результате мы получим другое изображение только с белыми и чёрными пикселями. Изображение изначально может быть цветным, т.е иметь три канала цвета: красный, зелёный, синий, поэтому необходимо перевести его в градации серого с помощью метода *cv.cvtColor* для дальнейшего анализа.

Какие пиксели будут белыми, а какие чёрными определяется с помощью порогового значения яркости пикселей. Значение яркости каждого отдельного пикселя может варьироваться от 0 до 255, где 0 – чёрный, а 255 – белый. Так как OpenCV работает с белыми объектами на чёрном фоне, нам необходимо сделать пиксели объектов, которые предположительно будут содержать итоговый результат работы программы, белыми, а все остальные – чёрными [10].

$$e_{ij} = \begin{cases} 255, e_{ij} = tv \\ 0, e_{ij} < tv \end{cases}, \quad (7)$$

где e_{ij} – элемент i-ой строки и j-ого столбца;

tv – пороговое значение.

Для наших целей пороговое значение устанавливается равное 255, так как всё равно в дальнейшем мы будем выделять и проверять все получившиеся контуры. Это означает, что для всех пикселей со значением яркости, равным 255, будет устанавливаться значение яркости, равное 0, а для всех остальных пикселей – равное 255. Сама фильтрация производится с помощью метода *cv.adaptiveThreshold*, её результат можно видеть на рисунке 2.2.

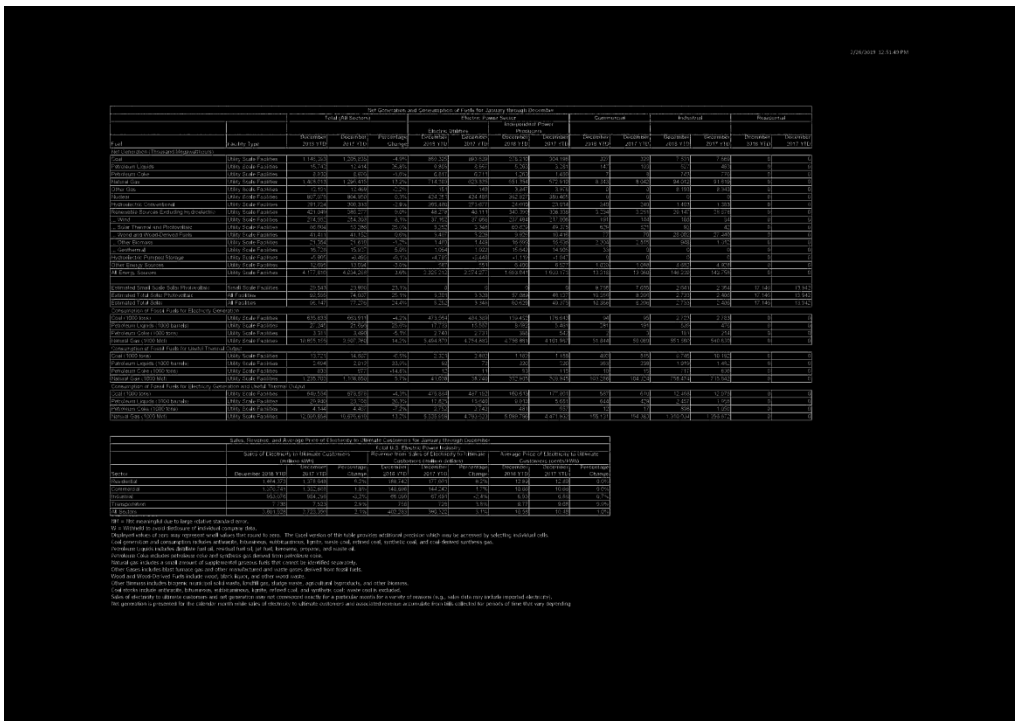

Рисунок 2.2 – Результат фильрации изображения.

В итоге будет получено двоичное изображение. Это делается для возможности осуществления изоляции линий.

Под изоляцией линий подразумевается морфологическое преобразование, основанное на форме объекта на изображении [10]. Такое преобразование проводится из-за возможности несоответствия объектов на изображении их предполагаемой форме, т.е. имеются пиксели, которые нарушают структурную целостность формы объекта, например, как показано на рисунке 2.3.

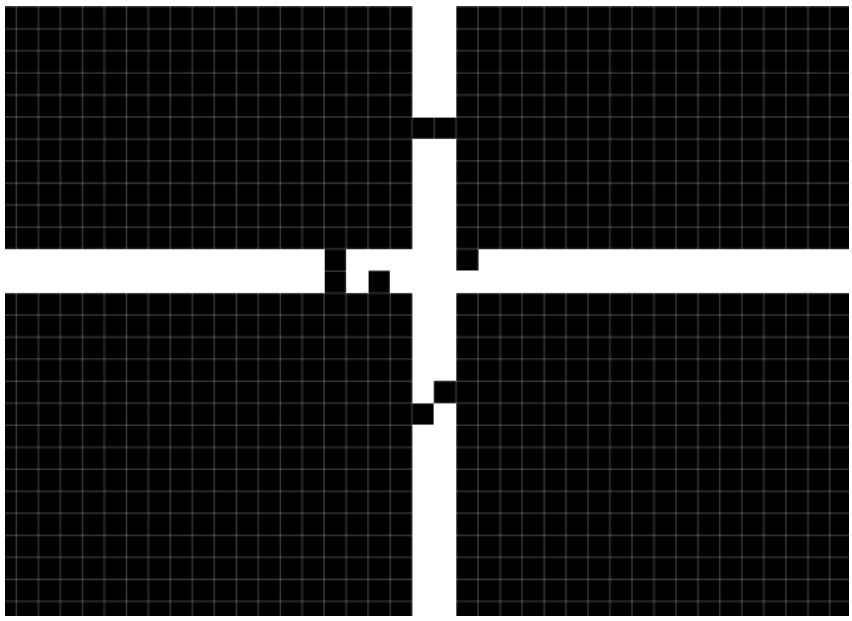

Рисунок 2.3 – Пример нарушения целостности объекта.

Видно, что некоторые пиксели нарушают прямоугольную форму объекта. Чтобы исправить ситуацию, необходимо задать цвет смежных им пикселей белым, т.е. $e_{ij} = 255$. Для наглядности пример результата работы морфологического преобразования приведён на рисунке 2.4.

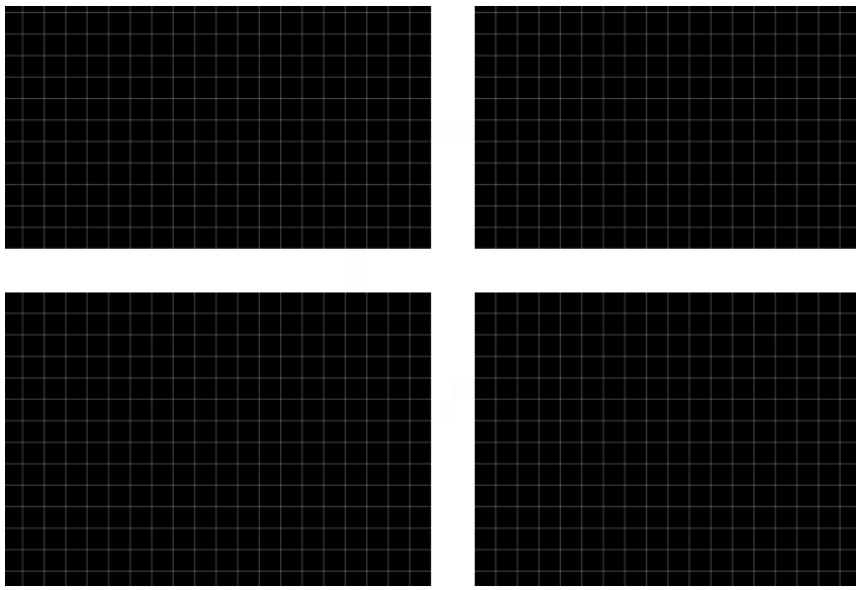

Рисунок 2.4 – Пример результата морфологического преобразования.

Преобразованию на вход подаются два параметра это исходное изображение и структурный элемент, который определяет характер преобразования. В данном случае необходимо взять прямоугольную форму

структурного элемента, так как линии на изображении фактически должны иметь именно такую форму.

В нашем случае для преобразования изображения будет использована функция *cv.getStructuringElement*, причём дважды: отдельно для горизонтальных и для верктикальных линий. Впоследствие будет найдено пересечение двух полученных изображений. На вход функции передаётся параметр *cv.MORPH_RECT*, означающий, что структурирующий элемент будет прямоугольной формы.

2.2.2 Выделение таблиц

Прежде чем непосредственно выделять таблицы, необходимо создать маску как сумму горизонтальных и вертикальных линий, найти все её контуры с помощью функции *cv.findContours* и найти пересечение горизонтальных и вертикальных линий [14]. Это производится для того, чтобы осуществить проверку на то, что выделенные контуры действительно являются таблицами. Эта проверка включает в себя следующее:

- Проверка на минимально возможное количество пикселей в области контура – 50;
- Аппроксимацию контура с помощью алгоритма Рамера-Дугласа-Пекера;
- Нахождение ограничивающего прямоугольника набора точек аппроксимации;
- Нахождение количества точек соединения в каждом регионе пересечения;
- Проверка на минимально возможное количество точек соединения – 5.

Если контур проходит проверку, то в коде программы для него создаётся новый экземпляр таблицы с координатами ограничивающего прямугольника набора точек аппроксимации. Затем для этого экземпляра

задаются полученные ранее координаты точек соединения, отсортированные от последнего к первому.

2.2.3 Распознавание текста и запись в файл

Для распознавания текста на конкретном языке Tesseract использует языковые модели и словари. Языковая модель содержит в себе значения параметров модели нейронной сети и другие данные обучения [15]. Прежде чем применять такую модель на практике, необходимо сначала обучить её, используя множество примеров как печатного текста, так и рукописного. Для этого случая существует база *EMNIST* с огромным количеством примеров печатных и рукописных символов, созданная специально для обучения моделей для распознавания текста.

Перед обучением все тренировочные изображения должны быть конвертированы в формат *tif* и объединены в многостраничный файл. Для того, чтобы выполнить обучение, нужно выполнить следующие команды в командной строке:

tesseract teslang.font.exp.tif teslang.font.exp nobatch box.train

unicharset_extractor teslang.font.exp.box

echo font 0 0 0 0 0>font_properties

mftraining -F font_properties -U teslang.unicharset -O teslang.unicharset teslang.font.exp.tr

cntraining teslang.font.exp.tr

combine_tessdata teslang

Таким образом, будет создана новая модель данных в Tesseract с именем *teslang.traineddata*, при распознавании текста будет использована именно эта модель. Для применения модели будем использовать заранее определённую функцию *run_tesseract* в отдельном файле:

```
def run_tesseract(filename, img_id, psm, oem):
    mkdir("bin/extracted/")
```

```
image = Image.open(filename)
language = 'teslang'
configuration = "--psm " + str(psm) + " --oem " + str(oem)
text = tess.image_to_string(image, lang=language, config=configuration)
if len(text.strip()) == 0:
    configuration += " -c tessedit_char_whitelist=0123456789"
    text = tess.image_to_string(image, lang=language, config=configuration)
return text
```

Результаты распознавания представлены на рисунке 2.5.

		Net Generation and Consumption of Fuels for December				
		Total (All Sectors)		Electric Power Sector		
				Electric Utilities	Independent Power Producers	
Fuel	Facility Type	December 2018	December 2017	Percentage Change	December 2018	December 2017
Net Generation (Thousand Megawatthours)						
Coal	Utility Scale Facilities	96825	106546	-9,10%		
Petroleum Liquids	Utility Scale Facilities	930	1982	-53,10%		
Petroleum Coke	Utility Scale Facilities	807	737	9,50%		
Natural Gas	Utility Scale Facilities	106978	111373	-3,90%		
Other Gas	Utility Scale Facilities	998	1096	-9,00%		
Nuclear	Utility Scale Facilities	71657	73700	-2,80%		
Hydroelectric Conventional	Utility Scale Facilities	23728	22377	6,00%		
Renewable Sources Excluding Hydroelectric	Utility Scale Facilities	34787	35151	-1,00%		
… Wind	Utility Scale Facilities	24825	24575	1,00%		
… Solar Thermal and Photovoltaic	Utility Scale Facilities	3188	3389	-5,90%		
… Wood and Wood-Derived Fuels	Utility Scale Facilities	3414	3738	-8,70%		
… Other Biomass	Utility Scale Facilities	1825	1877	-2,80%		
… Geothermal	Utility Scale Facilities	1535	1571	-2,30%		
Hydroelectric Pumped Storage	Utility Scale Facilities	-522	-656	-20,40%		
Other Energy Sources	Utility Scale Facilities	1147	1146	0,10%		
All Energy Sources	Utility Scale Facilities	337334	353452	-4,60%		
Estimated Small Scale Solar Photovoltaic	Small Scale Facilities	1774	1472	20,50%		
Estimated Total Solar Photovoltaic	All Facilities	4870	4739	2,80%		
Estimated Total Solar	All Facilities	4962	4861	2,10%		
Consumption of Fossil Fuels for Electricity Generation						

Рисунок 2.5 – Результат распознавания.

На рисунке 2.5 можно видеть, что результат распознавания довольно точный. Несмотря на то, что в некоторых строках присутствуют ошибочно распознанный символ «…», самый главный критерий – возможность более удобной работы с данными, выполнен.

2.3 Оценка точности

Для того, чтобы понимать насколько точный результат выдаёт программа, будем сравнивать полученный результат с простым ручнным вводом данных.

Для оценки точности при распознавании рассчитывалось общее количество символов $n_{\text{общ}}$ и количество ошибок $n_{\text{ош}}$. Затем выполняется расчёт точности распознавания по формуле

$$A = \frac{n_{\text{общ}} - n_{\text{ош}}}{n_{\text{общ}}} * 100\%. \quad (8)$$

Время распознавания одной страницы рассчитывалось по формуле

$$t_{\text{ст}} = \frac{t_{\text{общ}}}{N}, \quad (9)$$

где N – количество распознанных страниц; $t_{\text{общ}}$ – время распознавания всех страниц.

Для проведения исследования точности использовался ежемесячный финансовый отчёт об электроэнергии ПАО «Газпром» за декабрь 2018 года, при этом было проведено несколько попыток распознавания с постепенным увеличением количества страниц. Результаты проведения оценки точности представлены в таблице 1.

Таблица 1. Количественные характеристики качества распознавания программой

Количество распознанных страниц, N	Количество символов $n_{\text{общ}}$	Количество ошибок $n_{\text{ош}}$	Время распознавания одной страницы $t_{\text{ст}}$, с	Точность распознавания A, %
5	13 858	41	2.694	99.7
10	29 322	104	2.994	99.6
15	42 117	142	3,154	99.7
20	51 656	182	3.403	99.6
25	62 050	213	3,502	99.7

Исходя из полученных показателей можно сделать краткий вывод. Точность распознавания символов в таблицах в среднем составляет 99.7%, что является очень хорошим результатом, учитывая что чбор и обработка информации при распознавании – весьма ёмкий и длительный процесс. Это подчёркивается тем, что время распознавания одной страницы увеличивается с увеличением количества распознанных страниц. Подобные данные позволят анализировать и планировать временные затраты при больших объёмах распознаваемого материала.

3. Социальная ответственность

Целью магистерской диссертации является реализация алгоритма по распознаванию таблиц и текста, внесённого в них, на изображении. Данная работа предполагает использование ЭВМ для осуществления вычислений.

Рабочее место при написании магистерской работы укомплектовано следующим образом, стол компьютерный, офисное кресло кожаное подъёмно-поворотное и регулируемое по высоте и углам наклона сиденья и спинки, а также расстоянию спинки до переднего края сиденья, персональный компьютер со всеми необходимыми для работы периферийными устройствами, стереосистема, принтер, настольная лампа. Рабочее место находится в помещении, которое имеет следующие характеристики: ширина комнаты составляет $b = 3$м, длина $a = 5$м, высота H=2,7м. Тогда площадь помещения будет составлять $S = a \cdot b = 15\text{м}^2$, объем помещения $S = a \cdot b \cdot H = 40{,}5\text{м}^3$. В помещении имеется окно, через которое осуществляется вентиляция помещения. В помещении отсутствует принудительная вентиляция, т.е. воздух поступает и удаляется через дверь и окно, вентиляция является естественной.

В связи с тем, что целью исследования является распознавание информации в финансовых документах с помощью нейронных сетей, то основной целью является ускорение и автоматизация оценки вышеуказанных параметров. Областью применения данного исследования является использование машинного обучения в финансовых институтах и банках как способ автоматизации процесса распознавания информации.

3.1 Правовые и организационные вопросы обеспечения безопасности

3.1.1 Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства

По ТК РФ №197 – ФЗ работник имеет право на рабочее место, отвечающего требованиям охраны труда, обязательное соц. страхование от несчастных случаев и заболеваний, связанных с производством и профессией, получение информации от работодателя и государственных органов об условиях труда на рабочем месте и возможных рисках повреждения и утраты здоровья, а также о методах защиты и предотвращение опасных производственных факторов, а также:

- Право на отказ от выполнения работ при нарушении требований охраны труда, приводящих к возникновению опасности для жизни или здоровья работника кроме случаев, предусмотренных федеральными законами до ее устранения;
- Право на средства индивидуальной защиты во время проведения работ за счет работодателя; право на средства индивидуальной защиты во время проведения работ за счет работодателя;
- Право на обучение методам работы за счет работодателя.
- Право на прямое или косвенное участие в обсуждении вопросов, касающихся безопасных условий труда на рабочем месте и участие в рассмотрении произошедшего с ним несчастного случая или профессионального заболевания.
- Право на прохождение медицинского осмотра с сохранением средней ЗП и должности на время прохождения.
- Право на гарантии и компенсации, установленные договором при работе во вредных и опасных условиях, эти гарантии устанавливаются договором с учетом финансового положения работодателя.

3.1.2 Организационные мероприятия при компоновке рабочей зоны

При устройстве рабочего места человека, работающего за ПК необходимо соблюсти следующие основные условия: наилучшее местоположение оборудования и свободное рабочее пространство. Так как данная работа выполнена на персональном компьютере, в котором присутствуют все элементы стандартного ПК (системный блок, отдельный монитор, клавиатура и т.д.), то основным требованием к организации рабочего места является размещение монитора по центру письменного стола строго напротив пользователя, т.к. это обеспечивает положение монитора на уровне глаз оператора, а также комфортное положение рук оператора над клавиатурой.

При проектировании письменного стола должны быть учтены следующие требования согласно ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования» [25] и ГОСТ 12.2.061-81 «ССБТ. Оборудование производственное. Общие требования безопасности к рабочим местам» [26]. Высота рабочей поверхности стола рекомендуется в пределах 680–800 мм. Высота рабочей поверхности, на которую устанавливается клавиатура, должна быть 650 мм. Рабочий стол должен быть шириной не менее 700 мм и длиной не менее 1400 мм. Должно иметься пространство для ног высотой не менее 600 мм, шириной не менее 500 мм, глубиной на уровне колен не менее 450 мм и на уровне вытянутых ног не менее 650 мм.

Рабочее кресло должно быть подъёмно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки до переднего края сиденья. Рекомендуемая высота сидения над уровнем пола 420–550 мм. Конструкция рабочего кресла должна обеспечивать: ширину и глубину поверхности сиденья не менее 400 мм.

СанПиН 2.2.3670-20 «Санитарно-эпидемиологические требования к условиям труда»:

- яркость дисплея не должна быть слишком низкой или слишком высокой;
- размеры монитора и символов на дисплее должны быть оптимальными;
- цветовые параметры должны быть отрегулированы таким образом, чтобы не возникало утомления глаз и головной боли.
- опоры для рук не должны мешать работе на клавиатуре;
- верхний край монитора должен находиться на одном уровне с глазом, нижний – примерно на 20° ниже уровня глаза;
- дисплей должен находиться на расстоянии 45-60 см от глаз;
- локтевой сустав при работе с клавиатурой нужно держать под углом 90°;
- каждые 10 минут нужно отводить взгляд от дисплея примерно на 5-10 секунд;
- монитор должен иметь антибликовое покрытие;
- работа за компьютером не должна длиться более 6 часов, при этом необходимо каждые 2 часа делать перерывы по 15-20 минут;
- высота стола и рабочего кресла должны быть комфортными.

3.2 Производственная безопасность

3.2.1 Анализ вредных и опасных факторов, которые могут возникнуть на рабочем месте исследователя

Для идентификации потенциальных факторов использован ГОСТ 12.0.003-2015 «Опасные и вредные производственные факторы. Классификация». Перечень опасных и вредных факторов, характерных для проектируемой производственной среды представлен в виде таблицы 2.

Таблица 2. Возможные опасные и вредные факторы

Вредные факторы	Этапы работ			Нормативные документы
	Разработка	Изготовление	Эксплуатация	
1. Отклонение показателей микроклимата рабочей зоны.	+	+	+	СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений».
2. Недостаточная освещенность рабочей зоны.	+	+	+	СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий».
3. Отсутствие или недостаток естественного света.	+	+	+	СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий».
4. Повышенный уровень шума на рабочем месте.	+	+	+	Шум. Общие требования безопасности, СН 2.2.4/2.1.8.562–96
5. Повышенное образование электростатических зарядов.	+	+	+	СанПиН 2.2.2.542-96 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работ».

Анализ опасных и вредных производственных факторов

Отклонение показателей микроклимата рабочей зоны

Показатели микроклимата должны обеспечивать сохранение теплового баланса человека с окружающей средой и поддержание оптимального или допустимого теплового состояния организма.

Оптимальные микроклиматические при воздействии на человека в течение рабочей смены обеспечивают сохранение теплового состояния организма и не вызывают отклонений в состоянии здоровья. Допустимые микроклиматические условия могут приводить к незначительным дискомфортным тепловым ощущениям. Возможно временное (в течение рабочей смены) снижение работоспособности, без нарушения здоровья.

Все категории работ разграничиваются на основе интенсивности энергозатрат организма в ккал/ч (Вт). Работа, производимая сидя и сопровождающаяся незначительным физическим напряжением, относится к категории Iа – работа с интенсивностью энергозатрат до 120 ккал/ч (до 139 Вт). Для данной категории допустимые нормы микроклимата представлены в таблице 3.

Таблица 3. Допустимые нормы микроклимата в рабочей зоне производственных помещений

Период года	Категория тяжести выполняемых работ	Температура, °С		Относительная влажность, %		Скорость движения воздуха, м/с	
		Фактическое значение	Допустимое значение	Фактическое значение	Допустимое значение	Фактическое значение	Допустимое значение
Холодный	Iа	(20÷24)	(19÷24)	5	(15÷75)	0.1	≤ 0.1
Теплый	Iа	(23÷25)	(20÷28)	5	(15÷75)	0.1	≤ 0.2

Анализируя таблицу 3, можно сделать вывод, что в рассматриваемом помещении параметры микроклимата соответствуют нормам СанПиН. Допустимый уровень микроклимата помещения обеспечивается системой водяного центрального отопления и естественной вентиляцией.

Недостаточная освещенность рабочей зоны

Правильно спроектированное и выполненное освещение обеспечивает высокий уровень работоспособности, оказывает положительное психологическое действие на человека и способствует повышению производительности труда. На рабочей поверхности должны отсутствовать

резкие тени, которые создают неравномерное распределение поверхностей с различной яркостью в поле зрения, искажает размеры и формы объектов различия, в результате повышается утомляемость и снижается производительность труда.

В данном рабочем помещении используется комбинированное освещение: искусственное и естественное. Искусственное освещение создается люминесцентными лампами типа ЛД.

Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк согласно СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» [29].

Расчёт общего равномерного искусственного освещения горизонтальной рабочей поверхности выполняется методом коэффициента светового потока, учитывающим световой поток, отражённый от потолка и стен. Длина помещения $a = 5$м, ширина $b = 3$м, высота $H = 2{,}7$м. Высота рабочей поверхности над полом $h_p = 0{,}75$м. Интегральным критерием оптимальности расположения светильников является величина λ, которая для люминесцентных светильников с защитной решёткой лежит в диапазоне 1,1-1,3.

Выбираем лампу дневного света ЛД-40, световой поток которой равен $\Phi_{ЛД} = 2300$ Лм. Выбираем светильники с люминесцентными лампами типа ОДОР-2-30. Этот светильник имеет две лампы мощностью 40 Вт каждая, длина светильника равна 925 мм, ширина – 265 мм. На первом этапе определим значение индекса освещенности i.

$$i = \frac{S}{(a+b)h}, \tag{10}$$

где S – площадь помещения;

h – расчетная высота подвеса светильника, м;

a и b – длина и ширина помещения, м.

Высота светильника над рабочей поверхностью h

$$h = H - h_p - h_c = 2{,}7 - 0{,}75 - 0{,}3 = 1{,}65, \quad (11)$$

где H– высота помещения, м;

h_p– высота рабочей поверхности, м;

h_c– расстояние светильников от перекрытия (свес).

В результате проведенных расчетов, индекс освещенности i равен

$$i = \frac{S}{(a+b)\cdot h} = \frac{15}{(3+5)\cdot 1{,}65} = 1{,}14 \quad (12)$$

Расстояние между соседними светильниками равно 1,815 м или рядами определяется по формуле:

$$L = \lambda \cdot h = 1{,}1 \cdot 1{,}65 = 1{,}815 \quad (13)$$

Число рядов светильников в помещении:

$$Nb = \frac{b - \frac{2}{3}L}{L} + 1 = \frac{3 - \frac{2}{3} * 1{,}815}{1{,}815} + 1 = 1{,}98 \approx 2 \quad (14)$$

Число светильников в ряду равно 2:

$$Na = \frac{a - \frac{2}{3}L}{l_{\text{св}} + 0{,}5} = \frac{5 - \frac{2}{3} * 1{,}815}{0{,}925 + 0{,}5} = 2{,}66 \approx 3 \quad (15)$$

Общее число светильников:

$$N = Na \cdot Nb = 3 \cdot 2 = 6 \quad (16)$$

Учитывая, что в каждом светильнике установлено две лампы, общее число ламп в помещении $N = 12$.

Расстояние от крайних светильников или рядов до стены определяется по формуле:

$$l = \frac{L}{3} = \frac{1{,}815}{3} = 0{,}605 \quad (17)$$

Вышеуказанный расчет приведен в соответствии с нормами освещения рабочего места в помещении с вышеуказанными параметрами.

Потребный световой поток ламп в каждом из рядов:

$$\Phi = \frac{\text{Е}_\text{н} S \text{К}_\text{з} Z}{N\eta}, \quad (18)$$

Где $\text{Е}_\text{н}$ – нормируемая номинальная освещённость; S – площадь освещаемого освещения; $\text{К}_\text{з}$ – коэффициент запаса, учитывающий загрязнение

светильника; Z – коэффициент неравномерности освещения, для люминесцентных ламп он равен 1.1; N – число ламп в помещении; η – коэффициент использования светового потока.

Данное помещение относится к типу помещения со средним выделением пыли, поэтому коэффициент запаса равен 1.5. Состояние потолка – свежепобеленный, поэтому значение коэффициента отражения потолка $\rho_n = 70\%$. Состояние стен – бетонные стены, поэтому значение коэффициента отражения стен $\rho_c = 50\%$. $\rho_C = 50\%$. Коэффициент использования светового потока при $\rho_n = 70\%$ и $\rho_c = 50\%$ равен 0.43.

Подставляя значения в (18) получим:

$$\Phi = \frac{300 * 15 * 1{,}5 * 1{,}1}{12 * 0{,}43} = 1439 Лм$$

Для люминесцентных ламп с мощностью 40 ВТ и напряжением сети 220В, стандартный световой поток ЛД равен 2300 Лм. Проверяем выполнение условия:

$$-10\% \leq \frac{\Phi_{л.станд} - \Phi_{л.расч}}{\Phi_{л.станд}} * 100\% \leq +20\% \qquad (19)$$

Получаем: $-10\% \leq 37{,}4\% \leq +20\%$

Необходимый световой поток светильника выходит за пределы требуемого диапазона.

Определяем электрическую мощность осветительной установки

$$P = 12 * 40 = 480 \text{ Вт.}$$

Отсутствие или недостаток естественного света

Для оценки использования естественного света введено понятие коэффициента естественной освещенности (КЕО) и установлены минимальные допустимые значения КЕО – это отношение освещенности E_B внутри помещения за счет естественного света к наружной освещенности E_H от всей полусферы небосклона, выраженное в процентах:

$$КЕО = \frac{E_В}{E_Н} * 100\%.$$

При боковом естественном освещении в аудитории, лаборатории на рабочих столах и партах должен обеспечиваться КЕО = 1,5 %. К средствам нормализации освещения производственных помещений и рабочих мест относятся: источники света, осветительные приборы, световые проемы, светозащитные устройства, светофильтры

Повышенный уровень шума на рабочем месте

На компьютеризированных рабочих местах основными источниками шума являются вентиляторы системного блока, накопители, принтеры ударного действия. Шум создает значительную нагрузку на нервную систему человека, оказывая на него психологическое воздействие.

Уровень звука на рабочих местах, связанных с творческой деятельностью, научной деятельностью, программированием, преподаванием и обучением не должен превышать 50 дБА согласно [20].

Меры, которые необходимо принять, для того чтобы помещение было менее зашумленным – это обеспечить нормальную вентиляцию системного блока. Для охлаждения необходимо оборудовать со стороны вентиляционных отверстий 20-30 см свободного пространства.

Повышенное образование электростатических зарядов

Электризация заключается в следующем: нейтральные тела, в нормальном состоянии не проявляющие электрических свойств, при условии отрицательных контактов или взаимодействий становятся электрозаряженными. Опасность возникновения статического электричества проявляется в возможности образования электрической искры и вредном воздействии его на человеческий организм, и не только в случае непосредственного контакта с зарядом, но и за счет действий электрического

поля, которое возникает при заряде. При включенном питании компьютера на экране дисплея накапливается статическое электричество. Электрический ток искрового разряда статического электричества мал и не может вызвать поражение человека. Требования [30]:

1. Напряженность электромагнитного поля на расстоянии 50 см вокруг ВДТ по электрической составляющей должна быть не более:
   - в диапазоне частот 5Гц-2кГц - 25В/м;
   - в диапазоне частот 2кГц/400кГц - 2,5В/м.
2. Плотность магнитного потока должна быть не более:
   - в диапазоне частот 5Гц-2кГц - 250нТл;
   - в диапазоне частот 2кГц/400кГц - 25нТл.

При написании магистерской диссертации использовались следующие способы защиты от опасного воздействия электромагнитного излучения:

1. Защита временем (работа за компьютером осуществлялась не более двух часов подряд с 15 минутными перерывами);
2. Защита расстоянием (работа от экрана осуществлялась не менее 500 мм).

3.2.2 Обоснование мероприятий по защите персонала предприятия от действия опасных и вредных факторов

В ГОСТ 12.1.019-2009 приведены способы и средства защиты, обеспечивающие электробезопасность электроустановок различного назначения.

Согласно данному стандарту, для обеспечения защиты от случайного прикосновения к токоведущим частям, необходимо применять следующие способы и средства технической электробезопасности: защитные оболочки и ограждения, безопасное расположение токоведущих частей, изоляцию

токоведущий частей или рабочего места, малое напряжение и защитное отключение.

Для обеспечения защиты от поражения электрическим током при прикосновении к металлическим нетоковедущим частям, которые могут оказаться под напряжением в результате повреждения изоляции, применяют следующие способы: защитное заземление, зануление, выравнивание потенциала, систему защитных проводов, защитное отключение, изоляцию нетоковедущих частей, электрическое разделение сети, малое напряжение, контроль изоляции, компенсацию токов замыкания на землю, средства индивидуальной защиты. Описанные способы и средства применяют раздельно или в сочетании друг с другом так, чтобы обеспечивалась оптимальная защита.

К работе с электроустановками допускаются лица, прошедшие инструктаж и обучение безопасным методам труда, проверку знаний правил безопасности и инструкций в соответствии с занимаемой должностью с присвоением соответствующей квалификационной группы по технике безопасности и не имеющие медицинских противопоказаний.

3.3 Экологическая безопасность

Основным источником вреда для окружающей среды являются отходы, полученные при проведении работ. Их необходимо утилизировать. Так как израсходованная бумага не содержала никаких закрытых сведений, она была направлена на утилизацию без использования шредера, а люминесцентные лампы собраны и направлены на утилизацию в соответствующую организацию. Израсходованные картриджи аналогично были переданы производителю для централизованной утилизации в соответствии с требованиями ГОСТ 30775-2001.

Бытовой мусор помещений организаций несортированный, образованный в результате деятельности работников предприятия (код

отхода 91200400 01 00 4). Агрегатное состояние отхода твердое; основные компоненты: бумага и древесина, металлы, пластмассы и др. Для сбора мусора рабочее место оснащается урной. При заполнении урны, мусор выносится в контейнер бытовых отходов. Предприятие заключает договор с коммунальным хозяйством по вывозу и размещению мусора на организованных свалках.

Современное, электронное оборудование, даже бытовые приборы, нередко содержит драгметаллы, токсичные и прочие опасные для экологии и здоровья человека вещества. Фактически, порядок утилизации (избавления от) компьютеров и оргтехники производится с учетом двух федеральных законов:

N 89-ФЗ от 24.06.1998 – «Об отходах производства и потребления»;

N 41-ФЗ от 26.03.1998 – «О драгоценных металлах и драгоценных камнях».

Утилизация компьютерного оборудования осуществляется по специально разработанной схеме, которая должна соблюдаться в организациях На первом этапе необходимо создать комиссию, задача которой заключается в принятии решений по списанию морально устаревшей или не рабочей техники, каждый образец рассматривается с технической точки зрения;

Разрабатывается приказ о списании устройств. Для проведения экспертизы привлекается квалифицированное стороннее лицо или организация;

Составляется акт утилизации, основанного на результатах технического анализа, который подтверждает негодность оборудования для дальнейшего применения;

Формируется приказ на утилизацию. Все сопутствующие расходы должны отображаться в бухгалтерии;

Утилизацию оргтехники обязательно должна осуществлять специализированная фирма;

Получается специальная официальной формы, которая подтвердит успешность уничтожения электронного мусора.

После оформления всех необходимых документов, компьютерная техника вывозится со склада на перерабатывающую фабрику. Все полученные в ходе переработки материалы вторично используются в различных производственных процессах (ГОСТ 30494-2011, Здания жилые и общественные. Параметры микроклимата в помещениях, 2011.

3.4 Безопасность в чрезвычайных ситуациях

3.4.1 Анализ вероятных ЧС, которые может инициировать объект исследований

Так как данная работа выполняется на компьютере, то наиболее вероятной ЧС является возникновение пожара в помещении. Помещение, в котором велась работа по степени пожаробезопасности относится к категории Г (умеренная пожароопасность), т.е. к помещению, в котором находятся негорючие вещества и материалы в горячем, раскаленном или расплавленном состоянии, процесс обработки которых сопровождается выделением лучистого тепла, искр и пламени (СП 12.13130.2009).

Возникновение пожара может возникнуть от следующих источников воспламенения:

1) искра при разряде статистического электричества;
2) искра от электрооборудования;

Также на рабочем месте запрещается иметь огнеопасные вещества и выполнять следующее действия [32]:

1) курить;
2) зажигать огонь;
3) включать электрооборудование, если в помещении пахнет газом;
4) сушить что-либо на отопительных приборах;
5) закрывать вентиляционные отверстия в электроаппаратуре.

3.4.2 Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС

Согласно общим требованиям пожарной безопасности по ГОСТ 12.1.004-91 для устранения причин возникновения пожаров в помещении должны проводиться следующие мероприятия:

1. использование только исправного оборудования;
2. проведение периодических инструктажей по пожарной безопасности;
3. назначение ответственного за пожарную безопасность помещений;
4. издание приказов по вопросам усиления пожарной безопасности;
5. отключение электрооборудования, освещения и электропитания по окончании работ;
6. курение в строго отведенном месте;
7. содержание путей и проходов для эвакуации людей в свободном состоянии.

Для локализации или ликвидации возгорания на начальной стадии используются первичные средства пожаротушения. Первичные средства пожаротушения обычно применяют до прибытия пожарной команды.

Воздушно-пенные огнетушители очагов пожара, без наличия электроэнергии. Углекислотные и порошковые огнетушители предназначены для тушения электроустановок, находящихся под напряжением до 1000 В. Кроме того, порошковые применяют для тушения документов.

Для тушения токоведущих частей и электроустановок применяется переносной порошковый закачной огнетушитель ОП-3. Тушение электроустановок нужно производить на расстоянии не менее 1 метра (имеется в виду расстояние от сопла огнетушителя до токоведущих частей). Зарядку порошковых огнетушителей следует производить один раз в пять

лет. При возникновении необходимости ремонта или зарядки, следует обращаться в специализированные фирмы.

3.5 Выводы и рекомендации

Проанализировав условия труда на рабочем месте, где была разработана работа, можно сделать вывод, что помещение удовлетворяет необходимым нормам и в случае соблюдения техники безопасности и правил пользования компьютером работа в данном помещении не приведет к ухудшению здоровья работника.

Само помещение и рабочее место в нем удовлетворяет всем нормативным требованиям. Кроме того, действие вредных и опасных факторов сведено к минимуму, т.е. микроклимат, освещение и электробезопасность соответствуют требованиям, предъявленным в соответствующих нормативных документах.

Относительно рассмотренного вопроса об экологической безопасности можно сказать, что деятельность помещения не представляет опасности окружающей среде.

Важно добавить, что монитор компьютера служит источником ЭМП – вредного фактора, который отрицательно влияет на здоровье работника при продолжительной непрерывной работе и приводит к снижению работоспособности. Поэтому во избежание негативного влияния на здоровье необходимо делать перерывы при работе с ЭВМ и проводить специализированные комплексы упражнений для глаз.

4. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

Целью магистерской диссертации является реализация алгоритма по распознаванию таблиц и текста, внесённого в них, на изображении.

Целью раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является определения перспективности разработанного алгоритма распознавания таблиц и текста в них, успешности его реализации на рынке.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Определить потенциальных потребителей результатов исследования.

2. Выявить сильные и слабые стороны научно-исследовательского проекта, а также его возможности и вероятные угрозы при помощи SWOT-анализа.

3. Оценить степень готовности научного проекта к коммерциализации.

4. Определить заинтересованные стороны и ограничения/допущения научно-технического исследования; сформулировать цель и ожидаемые результаты проекта.

5. Определить структуру и трудоемкость выполнения работ, разработать график проведения научного исследования.

6. Рассчитать бюджет научно-технического исследования.

7. Определить риск возникновения неопределённых событий при выполнении НТИ, которые могут повлечь за собой нежелательные эффекты.

4.1 Потенциальные потребители результатов исследования

Для анализа потребителей результатов исследования необходимо рассмотреть целевой рынок и провести его сегментирование.

Целевой рынок – сегменты рынка, на котором будет продаваться в будущем разработка. В свою очередь, сегмент рынка – это особым образом выделенная часть рынка, группы потребителей, обладающих определенными общими признаками.

Сегментирование – это разделение покупателей на однородные группы, для каждой из которых может потребоваться определенный товар (услуга). Потенциальные потребители результатов исследования: банки, консалтинговые компании, финансовые холдинги, частные инвесторы.

4.2 Анализ конкурентных технических решений

Детальный анализ конкурирующих разработок, существующих на рынке необходимо проводить систематически, поскольку рынки пребывают в постоянном движении. Такой анализ помогает вносить коррективы в научное исследование, чтобы успешнее противостоять своим соперникам. Важно реалистично оценить сильные и слабые стороны разработок конкурентов.

С этой целью может быть использована вся имеющаяся информация о конкурентных разработках:

- технические характеристики разработки;
- конкурентоспособность разработки;
- уровень завершенности научного исследования (наличие макета, прототипа и т.п.);
- бюджет разработки;
- уровень проникновения на рынок;
- финансовое положение конкурентов, тенденции его изменения и т.д.

Анализ конкурентных технических решений с позиции ресурсоэффективности и ресурсосбережения позволяет провести оценку сравнительной эффективности научной разработки и определить направления для ее будущего повышения.

В ходе проведения исследований разрабатывается алгоритм распознавания с помощью нейронных сетей (обозначим через ф). Распознавать информацию с изображения можно и без нейронных сетей, поэтому в качестве конкурентов используем распознавание с помощью компьютерного зрения (конкурент к1) и использование ручного ввода информации (конкурент к2). Составим оценочную карту для сравнения конкурентных технических решений (таблица 4).

Таблица 4. Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		$Б_ф$	$Б_{к1}$	$Б_{к2}$	$К_ф$	$К_{к1}$	$К_{к2}$
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Повышение производительности труда пользователя (увеличение скорости расчёта, возможность работать с большими объемами данных)	0,09	5	4	3	0.41	0.33	0.25
2. Удобство в эксплуатации (соответствует требованиям потребителей)	0,09	4	4	1	0.44	0.44	0.11
3. Точность вычислений	0,1	4	4	3	0.36	0.36	0.27
4. Сложность вычислений	0,08	5	4	2	0.45	0.36	0.18
5. Доступность и простота (удобный формат, возможность вывода промежуточного результата и пр.) получаемых результатов	0,1	5	5	2	0.41	0.41	0.16
6. Адекватность модели и корректность результатов	0,1	4	4	3	0.36	0.36	0.27

Экономические критерии оценки эффективности							
1. Конкурентоспособность продукта	0,09	5	5	3	0.38	0.38	0.23
2. Уровень проникновения на рынок (степень внедрения данного продукта/услуги)	0,08	3	3	5	0.27	0.27	0.45
3. Стоимость продукта/услуги	0,1	5	5	1	0.45	0.45	0.09
4. Послепродажное обслуживание (техническая поддержка программного продукта/оказание дополнительных консультационных услуг)	0,1	5	5	1	0.45	0.45	0.09
5. Срок выхода на рынок	0,07	4	4	5	0.31	0.31	0.38
Итого	1	49	47	29	4.29	4.12	2.48

Критерии для сравнения и оценки ресурсоэффективности и ресурсосбережения, приведенные в таблице 4, подбираются исходя из выбранных объектов сравнения с учетом их технических и экономических особенностей разработки, создания и эксплуатации.

Позиция разработки и конкурентов оценивается по каждому показателю экспертным путем по пятибалльной шкале, где 1 – наиболее слабая позиция, а 5 – наиболее сильная. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1.

Анализ конкурентных технических решений определяется по формуле:

$$К = \sum В_i * Б_i, \quad (20)$$

где $К$ – конкурентоспособность научной разработки или конкурента;

$В_i$ – вес показателя (в долях единицы);

$Б_i$ – балл i-го показателя.

Основываясь на знаниях о конкурентах, можно объяснить, что, не смотря на простоту, простой ручной ввод, имеет более низкую скорость работы пользователя, особенно с большими массивами данных. Компьютерное зрение хотя и позволяет в некоторых случаях довольно точно

обработать информацию, но имеет серьёзный недостаток – неспособность системы «подстраиваться» под исходные данные, а значит не учитывает исторические данные.

4.3 SWOT-анализ

SWOT – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы) – представляет собой комплексный анализ научно-исследовательского проекта. SWOT-анализ применяют для исследования внешней и внутренней среды проекта.

Разработанная для данного исследования матрица SWOT представлена в таблице 5.

Таблица 5. SWOT-анализ

	Сильные стороны: С1. Высокая эффективность алгоритма для решения задач распознавания финансовой информации; С2. Гибкость алгоритма с точки зрения считывания исходных данных; С3. Простая эксплуатация.	**Слабые стороны**: Сл1. Узкий круг целевой аудитории; Сл2. Трудоемкий процесс написания и отладки программы для решения задач распознавания финансовой информации.
Возможности: В1. Расширение функционала; В2. Написание алгоритма на других языках программирования.	Благодаря гибкости и высокой эффективности алгоритм может быть дополнен реализацией других задач, например, распознавание графиков и диаграмм.	Написание алгоритма на другом языке программирования, а также расширение функционала могут сделать алгоритм более широко применимым, увеличить целевую аудиторию.
Угрозы: У1. Отсутствие спроса на продукт на рынке; У2. Развитие и появление аналогов алгоритма.	Наглядные результаты использования алгоритма, а также его низкая стоимость могут увеличить спрос на него.	Узкая направленность и использование платного языка программирования могут ослабить интерес покупателей.

4.4 Инициация проекта

Группа процессов инициации состоит из процессов, которые выполняются для определения нового проекта или новой фазы существующего. В рамках процессов инициации определяются изначальные цели и содержание и фиксируются изначальные финансовые ресурсы. Определяются внутренние и внешние заинтересованные стороны проекта, которые будут взаимодействовать и влиять на общий результат научного проекта. Данная информация закрепляется в уставе проекта.

Устав проекта документирует бизнес-потребности, текущее понимание потребностей заказчика проекта, а также новый продукт, услугу или результат, который планируется создать. Устав научного проекта магистерской работы должен иметь следующую структуру.

1. *Цели и результат проекта*

Под заинтересованными сторонами проекта понимаются лица или организации, которые активно участвуют в проекте или интересы которых могут быть затронуты как положительно, так и отрицательно в ходе исполнения или в результате завершения проекта. Информация по заинтересованным сторонам проекта представлена в таблице 6.

Таблица 6. Заинтересованные стороны проекта

Заинтересованные стороны проекта	**Ожидания заинтересованных сторон**
НИ ТПУ, ОЭФ	Проведение исследований по данной теме с целью использования настоящей разработки в образовательных целях, а также использование данной разработки в качестве основы под иные проекты, выполняемые на базе ОЭФ.
Банки, финансовые холдинги, частные инвесторы, трейдеры	Использование данной разработки для распознавания финансовых данных с целью их последующего использования.

Таблица 7. Цели и результаты проекта

Цели проекта:	Распознавание финансовой информации с изображения.
Ожидаемые результаты проекта:	Получение обучающейся нейронной сети на основе исторических данных.

Критерии приемки результата проекта:	Получение финансовых данных с помощью нейронных сетей.
Требования к результату проекта:	Корректность результатов и их соответствие реальным. Результаты должны быть представлены в доступной форме.

2. *Организационная структура проекта*

На данном этапе работы необходимо решить следующие вопросы: кто будет входить в рабочую группу данного проекта, определить роль каждого участника в данном проекте, а также прописать функции, выполняемые каждым из участников и их трудозатраты в проекте.

3. *Ограничения и допущения проекта*

Ограничения проекта – это все факторы, сдерживающие свободу участников команды в работе над данным проектом (таблица 8).

Таблица 8. Ограничения проекта

Фактор	**Ограничения/ допущения**
Источник финансирования	НИ ТПУ, ОЭФ
Бюджет затрат НТИ	262000
Сроки проекта	4 месяца
Дата утверждения плана управления проектом	1.02.2021
Дата завершения проекта	30.05.2021

4.5 Определение трудоемкости работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, т.к. зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости используется следующая формула:

$$t_{\text{ож}} = \frac{3t_{min}+2t_{max}}{5}, \quad (21)$$

где t_{min} – предположительно минимальная продолжительность этапа в рабочих днях, определяемая методом экспертной оценки, t_{max} – предположительно максимальная продолжительность этапа в рабочих днях, определяемая методом экспертной оценки.

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях Т_p, учитывающая параллельность выполнения работ несколькими исполнителями.

$$T_p^i = \frac{t_{\text{ож}}^i}{\text{Ч}_i}, \quad (22)$$

где T_p^i – продолжительность одной работы, раб. дн., $t_{\text{ож}}^i$– ожидаемая трудоемкость выполнения одной работы, чел.-дн, Ч_i – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

Продолжительность каждого этапа рассчитывается по формуле:

$$t_{\text{раб}} = t_{\text{ож}} * k_{\text{д}}, \quad (23)$$

где $t_{\text{раб}}$ – длительность этапов в рабочих днях, $k_{\text{д}}$ – коэффициент, учитывающий дополнительное время на консультации и согласование работ ($k_{\text{д}} = 1.2$). Данные по трудозатратам представлены в таблице 9.

Таблица 9. Трудозатраты участников разработки проекта

№ п/п	ФИО участника проекта	Роль в проекте	Функции	Трудозатраты, час
1	Крицкий О. Л.	Научный руководитель проекта	1. Обеспечение проекта ресурсами со стороны исполнителя. 2. Руководство и контроль за выполнением работ. 3. Регулярный анализ хода выполнения работ.	33
2	Кулигин С. М.	Магистрант	1. Выполнение	558

			расчетов по проекту. 2. Подготовка отчета о проделанной работе.	
				Итого: 591

Календарный план проекта за вычетом выходных и праздничных дней для сотрудников, работающих по шестидневной рабочей неделе, представлен в таблице 10.

Таблица 10. Календарный план проекта

Код работы	Название	Длительность, дни	Дата начала работы	Дата окончания работы	Состав участников (ФИО исполнителей)
1	Составление и утверждение ТЗ	6	1.02.2021	7.02.2021	Крицкий О. Л. Кулигин С. М.
2	Подбор и изучение материалов по теме	16	8.02.2021	28.02.2021	Кулигин С. М.
3	Сбор и анализ исходных данных	4	1.03.2021	5.03.2021	Кулигин С. М.
4	Выбор метода выполнения работы	11	6.03.2021	20.03.2021	Кулигин С. М.
5	Календарное планирование работ по теме	7	21.03.2021	29.03.2021	Крицкий О. Л. Кулигин С. М.
6	Применение выбранного метода к данным	17	30.03.2021	18.04.2021	Кулигин С. М.
7	Тестирование и анализ результатов работы	8	19.04.2021	29.04.2021	Кулигин С. М.
8	Исправление найденных ошибок, доработка модели	9	30.04.2021	14.05.2021	Крицкий О. Л. Кулигин С. М.
9	Составление пояснительной записки к магистерской диссертации	9	15.05.2021	25.05.2021	Кулигин С. М.
10	Оформление пояснительной записки к магистерской диссертации по	5	27.05.2021	30.05.2021	Кулигин С. М.

	ГОСТу				
Итого:		92			

Диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ (таблица 11).

Таблица 11. Календарный план-график выполнения работ

Номер	Наименование работы	Исполнители	Дни	Продолжительность выполнения работ, дни									
				Февраль		Март			Апрель		Май		
				7	20	5	14	8	20	10	14	10	6
1	Составление и утверждение ТЗ	М НР	7										
2	Подбор и изучение материалов по теме	М	20										
3	Сбор и анализ исходных данных	М	5										
4	Выбор метода выполнения работы	М	14										
5	Календарное планирование работ по теме	М НР	8										
6	Применение выбранного метода к данным	М	20										
7	Тестирование и анализ результатов работы	М	10										
8	Исправление найденных ошибок, доработка модели	М НР	14										
9	Составление пояснительной записки к магистерской диссертации	М	10										
10	Оформление пояснительной	М	5										

	записки к магистерской диссертации по ГОСТу												

Научный Руководитель (НР)

Магистрант (М)

4.6 Бюджет научно-технического исследования

При планировании бюджета НТИ должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета НТИ используется следующая группировка затрат по статьям:

1. материальные затраты НТИ;
2. затраты на специальное оборудование для научных (экспериментальных) работ;
3. основная заработная плата исполнителей темы;
4. дополнительная заработная плата исполнителей темы;
5. отчисления во внебюджетные фонды (страховые отчисления);
6. затраты научные и производственные командировки;
7. контрагентные расходы;
8. накладные расходы.

В нашем случает расчет основных затрат производится только для научного руководителя, так как исполнитель является студентом, и, следовательно, со стороны ТПУ него нет затрат.

4.6.1 Расчёт материальных затрат НТИ

Данная статья отражает стоимость всех материалов, используемых при разработке проекта, включая расходы на их приобретение и доставку.

Транспортные расходы принимаются в пределах 3-5% от стоимости материалов. В материальные затраты, помимо вышеуказанных, включаются дополнительно затраты на канцелярские принадлежности, диски, картриджи и т.п. Однако их учет ведется в данной статье только в том случае, если в научной организации их не включают в расходы на использование оборудования или накладные расходы. Расчёт затрат на материалы производится по форме, приведенной в таблице 12.

Таблица 12. Материальные затраты

Наименование	Единица измерения	Количество	Цена за ед., руб.	Затраты на материалы, руб.
Персональный компьютер (Ryzen 7/16ГБ ОЗУ)	Шт	1	71000	71000
Microsoft Windows 10 Professional RU x32/x64	Шт	1	9000	9000
Пакет Microsoft Office 2010 Home and Student RU x32/x64	Шт	1	4 600	4 600
Элекроэнергия	кВт	102,3	5,8	593,34
Канцелярские принадлежности	Шт	1	300	300
			Итого	85493,34

4.6.2 Основная заработная плата

Величина расходов по заработной плате определяется исходя из трудоемкости выполняемых работ и действующей системы окладов и тарифных ставок. В состав основной заработной платы включается премия, выплачиваемая ежемесячно из фонда заработной платы в размере 20-30 % от тарифа или оклада.

Статья включает основную заработную плату работников, непосредственно занятых выполнением НТИ, (включая премии, доплаты) и дополнительную заработную плату:

$$З_{зп} = З_{осн} + З_{доп}, \quad (24)$$

где $З_{ОСН}$ – основная заработная плата;

$З_{доп}$ – дополнительная заработная плата.

Основная заработная плата ($З_{ОСН}$) руководителя (лаборанта, инженера) от предприятия (при наличии руководителя от предприятия) рассчитывается по следующей формуле:

$$З_{осн} = З_{дн} \cdot Т_{р}, \quad (25)$$

где $З_{ОСН}$ – основная заработная плата;

$Т_{Р}$ – продолжительность работ, выполняемых научно-техническим работником, раб.дн.;

$З_{дн}$ – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$З_{дн} = \frac{З_{м} \cdot М}{F_{д}} \quad (26)$$

где $З_{М}$ – месячный должностной оклад работника, руб.;

М – количество месяцев работы без отпуска в течение года (при отпуске в 48 раб.дней М=10,4 месяца, 6-дневная неделя);

$F_{д}$ – действительный годовой фонд рабочего времени научно-технического персонала, раб.дн.

Месячный должностной оклад работника:

$$З_{м} = З_{тс} \cdot (1 + k_{пр} + k_{д}) \cdot k_{р}, \quad (27)$$

где $З_{ТС}$ – заработная плата по тарифной ставке, руб.;

$k_{пр}$ – премиальный коэффициент, равный 0,3 (т.е. 30% от Зтс);

$k_д$ – коэффициент доплат и надбавок составляет примерно 0,2-0,5 (в НИИ и на промышленных предприятиях – за расширение сфер обслуживания, за профессиональное мастерство, за вредные условия: 15-20 % от Зтс);

$k_р$ – районный коэффициент, равный 1,3 г. Томск.

Пример расчета заработной платы для руководителя:

$$З_м = З_{тс} \cdot (1 + k_{пр} + k_д) \cdot k_р = 33664 \cdot (1 + 0,3 + 0,2) \cdot 1,3 = 65664,8 руб.$$

$$З_{дн} = \frac{65664,8 \cdot 10,4}{251} = 2720 руб.$$

$$З_{осн} = З_{дн} \cdot Т_р = 2720 \cdot 5,5 = 14960 руб.$$

Таблица 13. Расчёт основной заработной платы

Исполнители	$З_{тс}$	$k_р$	$З_м$, руб	$З_{дн}$, руб	$Т_р$, дни	$З_{осн}$, руб
Научный Руководитель	33664	1,3	65664,8	2720	5,5	14960
Магистрант	12663	1,3	24634,5	1021	93	94953
					Итого	109913

4.6.3 Дополнительная заработная плата

Затраты по дополнительной заработной плате исполнителей темы учитывают величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций (при исполнении государственных и общественных обязанностей, при совмещении работы с обучением, при предоставлении ежегодного оплачиваемого отпуска и т.д.).

Расчет дополнительной заработной платы ведется по следующей формуле:

$$З_{доп} = k_{доп} \cdot З_{осн}, \quad (28)$$

где $k_{доп}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12).

Таблица 14. Расчет дополнительной заработной платы

Исполнители	Основная ЗП, руб	Дополнительная ЗП, руб
Руководитель	14960	1795
Магистрант	94953	11394
	Итого	13189

4.6.4 Отчисления во внебюджетные фонды

Отчисления во внебюджетные фонды являются обязательными по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$З_{внеб} = k_{внеб} \cdot (З_{осн} + З_{доп}), \quad (29)$$

где $k_{внеб}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

Установлен размер страховых взносов равный 30.2%. Отчисления во внебюджетные фонды представлены в таблице 15.

Таблица 15. Отчисления во внебюджетные фонды

Исполнители	Основная ЗП, руб	Сумма отчисления, руб
Руководитель	14960	4518
Магистрант	94953	28676
	Итого	33194

4.6.5 Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$З_{накл} = (З_{осн} + З_{доп}) \cdot k_{нр}, \quad (30)$$

где $k_{нр}$ – коэффициент, учитывающий накладные расходы (принимается равным 30%): $З_{накл} = 36930,6 руб.$

Затраты на электроэнергию, потребляемую персональным компьютером 220 Вт, при среднем времени работы 5 часа в день и 93 днях работы, а также при стоимости 1 кВт = 5,8 руб. получим:

$$З_{эл} = \text{мощность} \cdot \text{часы} \cdot \text{дни} \cdot \text{тариф}$$

$$З_{эл} = 0,22 \cdot 5 \cdot 93 \cdot 5,8 = 593,34$$

4.6.6 Формирование бюджета затрат НТИ

Рассчитанная величина затрат научно-исследовательской работы (темы) является основой для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку научно-технической продукции. Определение бюджета затрат на научно-исследовательский проект приведен в таблице 16.

Таблица 16. Расчет бюджета затрат НТИ

Наименование статьи	**Сумма, руб.**
1. Материальные затраты НТИ	51493,34
2. Затраты по основной заработной плате исполнителей темы	109913
3. Затраты по дополнительной заработной плате исполнителей темы	13189
4. Отчисления во внебюджетные фонды	29786
5. Накладные расходы	36930,6
6. Затраты на электроэнергию	593,34
8. Бюджет затрат НТИ	291965,28

4.7 Реестр рисков проекта

Идентифицированные риски проекта включают в себя возможные неопределенные события, которые могут возникнуть в проекте и вызвать

последствия, которые повлекут за собой нежелательные эффекты. Потенциальные риски представлены в таблице 17.

Таблица 17. Реестр рисков

№	Риск	Потенциальное воздействие	Вероятность наступления (1-5)	Влияние риска (1-5)	Уровень риска	Способы смягчения риска	Условия наступления
1	Ошибки в исторических данных	Получение заведомо неадекватного результата при корректно работающей программе	4	5	высокий	Использование достоверного источника данных	Использован недостоверный источник информации. Внесены ошибки в данные при переводе из одного формата в другой и пр.
2	Ошибки в программе (ошибки в алгоритме, ошибки/неточности в формулах)	Получение некорректных результатов или получение неработающей программы	2	4	средний	Доработка алгоритма, дебаг программы. Использование авторитетного источника информации, из которого берутся исходные формулы	Допущение ошибок при написании формул и составлении алгоритма. Использование непроверенного источника информации, следовательное, расчёт по некорректным формулам
3	Плохое техническое оснаще	Невозможность получения результатов	1	1	низкий	Установка ПО или использование	Отсутствие необходимого ПО у пользовате

	ние					иного ПК для работы	ля/несовместимость ПК с необходимым ПО

4.8 Оценка сравнительной эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности научного исследования получают в ходе оценки бюджета затрат трех (или более) вариантов исполнения научного исследования. Для этого наибольший интегральный показатель реализации технической задачи принимается за базу расчета (как знаменатель), с которым соотносится финансовые значения по всем вариантам исполнения.

Интегральный финансовый показатель разработки определяется как:

$$I_{финр}^{исп.i} = \frac{Ф_{pi}}{Ф_{\max}}, \quad (31)$$

где $I_{финр}^{исп.i}$ – интегральный финансовый показатель разработки;

$Ф_{pi}$ – стоимость *i*-го варианта исполнения;

$Ф_{\max}$ – максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги). За максимально возможную стоимость исполнения примем 300000 руб.

Полученная величина интегрального финансового показателя разработки отражает соответствующее численное увеличение бюджета затрат разработки в разах (значение больше единицы), либо соответствующее численное удешевление стоимости разработки в разах (значение меньше единицы, но больше нуля).

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum a_i \cdot b_i, \quad (32)$$

где I_{pi} – интегральный показатель ресурсоэффективности для i-го варианта исполнения разработки;

a_i – весовой коэффициент *i*-го варианта исполнения разработки;

a_i^a, b_i^p – бальная оценка *i*-го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;

n – число параметров сравнения.

Расчет интегрального показателя ресурсоэффективности представлен в таблице 18.

Таблица 18. Расчет интегрального показателя ресурсоэффективности

Критерии	Весовой коэффициент параметра	Исп.1: магистрант	Исп.2: конкурент
Способствует росту производительности труда пользователя	0,2	5	4
Возможность применения любым предприятием	0,15	3	4
Требует наличия исторических данных	0,25	5	5
Простота применения	0,15	4	5
Конкурентоспособность (с другими системами)	0,25	4	3
ИТОГО	1	4,3	4,15

$$I_{р–исп.1} = 5 \cdot 0,2 + 3 \cdot 0,15 + 5 \cdot 0,25 + 4 \cdot 0,15 + 4 \cdot 0,25 = 4,3$$

$$I_{р–исп.2} = 4 \cdot 0,2 + 4 \cdot 0,15 + 5 \cdot 0,25 + 5 \cdot 0,15 + 3 \cdot 0,25 = 4,15$$

$$I_{\max} = 4,3$$

Интегральный показатель эффективности вариантов исполнения разработки определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп.i} = \frac{I_{р-исп.i}}{I_{финр}^{исп.i}}, \quad (33)$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных. Сравнительная эффективность проекта определяется по формуле:

$$Э_{ср} = \frac{I_{исп.\text{i}}}{I_{исп.\max}}, \quad (34)$$

Сравнительная эффективность разработки представлена в таблице 19.

Таблица 19. Сравнительная эффективность разработки

№ п/п	Показатели	Проект магистранта	Проект конкурента
1	Интегральный финансовый показатель разработки	0,8	1
2	Интегральный показатель ресурсоэффективности разработки	4,3	4,15
3	Интегральный показатель эффективности	5,4	4,15
4	Сравнительная эффективность вариантов исполнения	1,25	0,96

Сравнение значений интегральных показателей эффективности позволяет понять и выбрать более эффективный вариант решения поставленной в магистерской работе технической задачи с позиции финансовой и ресурсной эффективности.

4.9 Оценка абсолютной эффективности исследования

В основе проектного подхода к инвестиционной деятельности предприятия лежит принцип денежных потоков (cashflow). Особенностью является его прогнозный и долгосрочный характер, поэтому в применяемом подходе к анализу учитываются фактор времени и фактор риска. Для оценки

общей экономической эффективности используются следующие основные показатели:

- чистая текущая стоимость (NPV);
- индекс доходности (PI);
- внутренняя ставка доходности (IRR);
- срок окупаемости (DPP).

Чистая текущая стоимость (NPV) – это показатель экономической эффективности инвестиционного проекта, который рассчитывается путём дисконтирования (приведения к текущей стоимости, т.е. на момент инвестирования) ожидаемых денежных потоков (как доходов, так и расходов).

Расчёт NPV осуществляется по следующей формуле:

$$\mathbf{NPV} = \sum_{\mathbf{t=1}}^{\mathbf{n}} \frac{\mathbf{ЧДП_{оп_t}}}{\mathbf{(1+i)^t}} - \mathbf{I_0} \qquad (35)$$

где: $ЧДП_{оп}{}_t$ – чистые денежные поступления от операционной деятельности;

I_0 – разовые инвестиции, осуществляемые в нулевом году;

t – номер шага расчета (t= 0, 1, 2 …n);

n – горизонт расчета;

i – ставка дисконтирования (желаемый уровень доходности инвестируемых средств).

Расчёт NPV позволяет судить о целесообразности инвестирования денежных средств. Если NPV>0, то проект оказывается эффективным.

Расчет чистой текущей стоимости представлен в таблице 19. При расчете рентабельность проекта составляла 30 %, норма амортизации – 10 %. Бюджет проекта=246965,28руб. стр.4(Операционные затраты)=Сырье+Амортизация +ФОТ(Осн.ЗП+доп.ЗП.соц.отч.) Вреал.=Бюджет (себестоимость)*1,2; Ц=С*(1+Р/100)

Таблица 20. Расчет чистой текущей стоимости по проекту в целом

№	Наименование показателей	Шаг расчета				
		0	1	2	3	4

1	Выручка от реализации, руб.	0	379554,86	379554,86	379554,86	379554,86
2	Итого приток,руб.	0	379554,86	379554,86	379554,86	379554,86
3	Инвестиционные издержки, руб.	-291965,28	0	0	0	0
4	Операционные затраты, руб.	0	254441,34	254441,34	254441,34	254441,34
5	Налогооблагаемая прибыль	0	125113,52	125113,52	125113,52	125113,52
6	Налоги 20 %, руб.	0	25022,7	25022,7	25022,7	25022,7
7	Итого отток, руб.	-291965,28	279464,04	279464,04	279464,04	279464,04
8	Чистая прибыль, руб.	0	100090,82	100090,82	100090,82	100090,82
9	Чистый денежный поток (ЧДП), руб.	-291965,28	150150,82	150150,82	150150,82	150150,82
10	Коэффициент дисконтирования (КД)	1	0,909	0,826	0,751	0,683
11	Чистый дисконтированный доход (ЧДД), руб.	-291965,28	136500,74	124091,59	112810,53	102555,03
12	$\sum$ЧДД	475957,89				
12	ИтогоNPV, руб.	183992,61				

Коэффициент дисконтирования рассчитан по формуле:

$$КД = \frac{1}{(1+i)^t} \quad (36)$$

где: –ставка дисконтирования, 10 %;

t – шаг расчета.

Таким образом, чистая текущая стоимость по проекту в целом составляет 183992,61 рублей, что позволяет судить об его эффективности.

Индекс доходности (PI) – показатель эффективности инвестиции, представляющий собой отношение дисконтированных доходов к размеру инвестиционного капитала. Данный показатель позволяет определить

инвестиционную эффективность вложений в данный проект. Индекс доходности рассчитывается по формуле:

$$PI = \sum_{t=1}^{n} \frac{ЧДП_t}{(1+i)^t} / I_0 \quad (37)$$

где: ЧДД - чистый денежный поток, руб.;

I_0– начальный инвестиционный капитал, руб.

PI=475957,89/291965,28=1,63

Так как PI>1, то проект является эффективным.

Значение ставки, при которой **NPV** обращается в нуль, носит название «внутренней ставки доходности» или IRR. Формальное определение «внутренней ставки доходности» заключается в том, что это та ставка дисконтирования, при которой суммы дисконтированных притоков денежных средств равны сумме дисконтированных оттоков или **NPV** =0. По разности между IRR и ставкой дисконтирования i можно судить о запасе экономической прочности инвестиционного проекта. Чем ближе IRR к ставке дисконтирования i, тем больше риск от инвестирования в данный проект.

$$\sum_{t=1}^{n} \frac{ЧДП_{оп_t}}{(1+IRR)^t} = \sum_{t=0}^{n} \frac{I_t}{(1+IRR)^t} \quad (38)$$

Между чистой текущей стоимостью (NPV) и ставкой дисконтирования (i) существует обратная зависимость. Эта зависимость представлена в таблице 20 и на рисунке 4.1.

Таблица 21. Зависимость NPVот ставки дисконтирования

№	Наименование показателя	0	1	2	3	4	NPV, руб.
1	Чистые денежные потоки, руб.	-291965,28	150150,82	150150,82	150150,82	150150,82	
2	Коэффициент дисконтирования						

	0,1	1	0,909	0,826	0,751	0,683	
	0,2	1	0,833	0,694	0,578	0,482	
	0,3	1	0,769	0,592	0,455	0,350	
	0,4	1	0,714	0,510	0,364	0,260	
	0,5	1	0,667	0,444	0,295	0,198	
	0,6	1	0,625	0,390	0,244	0,153	
	0,7	1	0,588	0,335	0,203	0,112	
	0,8	1	0,556	0,309	0,171	0,095	
	0,9	1	0,526	0,277	0,146	0,077	
	1	1	0,500	0,250	0,125	0,062	
3	Дисконтированный денежный доход, руб.						
	0,1	-291965,28	136500,74	124091,59	112810,53	102555,03	183992,61
	0,2	-291965,28	125125,68	104271,40	86892,84	72410,70	96735,33
	0,3	-291965,28	115500,63	88846,64	68343,57	52571,98	33297,53
	0,4	-291965,28	107250,59	76607,56	54719,69	39085,49	-14301,95
	0,5	-291965,28	100100,55	66733,70	44489,13	29659,42	-50982,48
	0,6	-291965,28	93844,26	58652,66	36657,91	22911,20	-79899,24
	0,7	-291965,28	88324,01	51955,31	30561,94	17977,61	-103146,41
	0,8	-291965,28	83417,12	46342,85	25746,03	14303,35	-122155,94
	0,9	-291965,28	79026,75	41593,02	21891,07	11521,61	-137932,83
	1,0	-291965,28	75075,41	37537,7	18768,85	9384,43	-151198,88

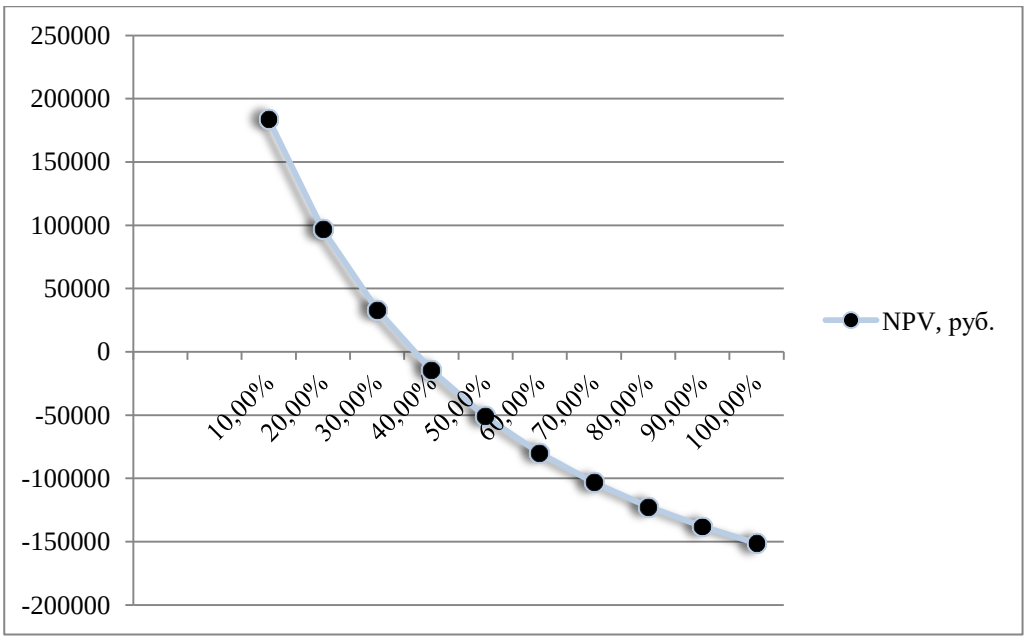

Рисунок 4.1. Зависимость NPV от ставки дисконтирования

Из таблицы 21 и рисунка 4.1 следует, что по мере роста ставки дисконтирования чистая текущая стоимость уменьшается, становясь отрицательной. Значение ставки, при которой NPV обращается в нуль, носит название «внутренней ставки доходности» или «внутренней нормы прибыли». Из графика получаем, что IRR составляет 0,38.

Запас экономической прочности проекта: 38%-10%=28%.

Как отмечалось ранее, одним из недостатков показателя простого срока окупаемости является игнорирование в процессе его расчета разной ценности денег во времени.

Этот недостаток устраняется путем определения дисконтированного срока окупаемости. То есть это время, за которое денежные средства должны совершить оборот.

Наиболее приемлемым методом установления дисконтированного срока окупаемости является расчет кумулятивного (нарастающим итогом) денежного потока (таблица 22).

Таблица 22. Дисконтированный срок окупаемости

№	Наименование показателя	Шаг расчета				
		0	1	2	3	4

1	Дисконтированный денежный доход ($\mathbf{i}$ =0,035), руб.	-291965,28	136500,74	124091,59	112810,53	102555,03
2	То же нарастающим итогом, руб.	-291965,28	-155464,54	-31375,95	81437,58	183992,61
3	Дисконтированный срок окупаемости	$\mathbf{PP_{дск}}$=2+(31375,95/112810,53)=2,65года				

Социальная эффективность научного проекта (таблица 23) учитывает социально-экономические последствия осуществления научного проекта для общества в целом или отдельных категорий населений или групп лиц, в том числе как непосредственные результаты проекта, так и «внешние» результаты в смежных секторах экономики: социальные, экологические и иные внеэкономические эффекты.

Таблица 23. Критерии социальной эффективности

ДО	ПОСЛЕ
Нерациональное использование временных ресурсов пользователя	Повышение производительности труда пользователя (увеличение скорости расчёта, возможность работать с большими объемами данных)
Потенциальные риски допущения ошибок при математическом расчете модели	Усовершенствование в использовании ПО, исключение риска потенциальных ошибок при математическом расчете модели

Социально-экономические последствия выражаются в нерациональности использования научного проекта на практике после его осуществления. Однако, с течением времени, проекта будет только улучшаться и рано или поздно достигнет той самой критической точки, когда он его использование станет целесообразным.

4.10 Выводы по главе «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

В ходе выполнения части работы по финансовому менеджменту, ресурсоэффективности и ресурсосбережению был проведен анализ разрабатываемого исследования.

Во-первых, оценен коммерческий потенциал и перспективность проведения исследования. Полученные результаты говорят о потенциале и перспективности на уровне выше среднего.

Во-вторых, проведён SWOT-анализ проекта, в ходе которого были выявлены потенциальные внутренние и внешние сильные и слабые стороны, возможности и угрозы. Из анализа выяснили, что сильные стороны, такие как высокая эффективность и гибкость алгоритма, преобладают над слабыми, поэтому отметим основные возможности проекта: расширение алгоритма и написание его на других языках программирования. Отсюда следует, что разработка программы, реализующий данный алгоритм, является перспективным проведением научного исследования.

В-третьих, проведено планирование НИР, а именно: определена структура и календарный план работы, трудоемкость и бюджет НТИ. Результаты соответствуют требованиям к магистерским диссертациям по срокам и иным параметрам. В ходе планирования научно-исследовательских работ определены структура и перечень работ, выполняемых рабочей группой. В данном случае рабочая группа состоит из руководителя и студента, длительность работ для руководителя составляет 22 дня, а для студента – 92 дня. Был построен календарный план-график на основе диаграммы Ганта, по которому можно увидеть, что самые продолжительные по времени работы – это подбор и изучение материалов по теме (20 дней) и применение выбранного метода к данным (20 дней).

В-четвёртых, определена эффективность исследования в разрезах ресурсной, финансовой, бюджетной, социальной и экономической эффективности.

В-пятых, Бюджет научно-технического исследования составил 291965,28 руб. Основную часть бюджета составила зарплата работников (109913 рублей).

Таким образом, капиталовложения в размере 291965,28 рубля позволят реализовать разработку алгоритма распознавания таблиц и текста в них. Алгоритм позволит существенно упростить и ускорить работу с финансовой информацией.

Заключение

В ходе выполнения были получены следующие результаты:

– сформулирована актуальность поставленной задачи;

– сформулированы концептуальная постановка задачи;

– сформулированы и обобщены общие этапы решения задачи;

– разработана модель алгоритма распознавания необходимой информации на заданном изображении;

– составлены алгоритм изоляции линий на изображении и алгоритм выделения таблиц;

– разработана программа на языке программирования Python, реализующая составленные алгоритмы.

– проверена и подтверждена работоспособность программы на тестовом примере;

– проведена оценка точности результата работы программы в сравнении с предполагаемым результатом.

Программа показала результат распознавания информации с точностью 99.7 % в среднем, что говорит о целесообразности использования программы на практике.

Список использованных источников

1. Клетте Р. Компьютерное зрение. Теория и алгоритмы / Р. Клетте, – М.: Изд-во «ДМК Пресс», 2019. – 506 с.
2. Шакирьянов Э. Д. Компьютерное зрение на Python. Первые шаги / Э. Д. Шакирьянов, – М.: Изд-во «Лаборатория знаний», 2021. – 160 с.
3. Солем Я. Э. Программирование компьютерного зрения на языке Python / Я. Э. Солем, – М.: Изд-во «ДМК Пресс», 2016. – 314 с.
4. Чару А. Нейронные сети и глубокое обучение. Учебный курс / А. Чару, – М.: Изд-во «Вильямс», 2020. – 752 с.
5. Understanding LSTM Networks [Электронный ресурс]. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (Дата обращения: 16.04.2021).
6. Алгоритм Дугласа-Пекера [Электронный ресурс]. URL: https://habr.com/ru/post/448618/ (Дата обращения: 23.03.2021).
7. Мюллер А. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными / А. Мюллер, С. Гвидо, – М.: Изд-во «Вильямс», 2017. – 480 с.
8. Лутц М. Изучаем Python / М. Лутц, – М.: Изд-во «Символ-Плюс», 2011. – 1280 с.
9. Франсуа Ш. Глубокое обучение на Python / Ш. Франсуа, – СПб.: Изд-во «Питер», 2018. – 400 с.
10. Вудс Р. Цифровая обработка изображений / Р. Вудс, Р. Гонсалес, – М.: Изд-во «Техносфера», 2012. – 1104 с.
11. Рашид Т. Создаём нейронную сеть / Т. Рашид. – М.: Изд-во «Вильямс», 2018. – 272 с.
12. Нейронные сети. Часть 1 [Электронный ресурс]. URL: https://habr.com/ru/post/312450/ (Дата обращения: 04.04.2021)

13. Пишем свою нейросеть: пошаговое руководство [Электронный ресурс]. URL: https://proglib.io/p/neural-nets-guide/ (Дата обращения: 30.03.2021).

14. Распознавание образов с использованием OpenCV [Электронный ресурс]. URL: http://recog.ru/wp-content/uploads/2020/05/opencvkruchinin.pdf (Дата обращения: 21.02.2021)

15. Tesseract User Manual [Электронный ресурс]. URL: https://github.com/tesseract-ocr/tessdoc#tesseract-user-manual (Дата обращения: 26.04.2021).

16. Финам.ru [электронный ресурс]. URL: https://www.finam.ru/ (дата обращения: 12.02.2020).

17. СанПиН 2.2.2/2.4.1.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

18. СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений».

19. СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий».

20. СанПиН 2.2.4/2.1.8.10-32-2002 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки».

21. СанПиН 2.2.2.542-96 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работ».

22. СанПиН 2.2.4.1191-03 «Электромагнитные поля в производственных условиях».

23. СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий».

24. СанПиН 2.2.4.3359-16 «Санитарно-эпидемиологические требования к физическим факторам на рабочих местах».

25. ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования».

26. ГОСТ 12.2.061-81 «ССБТ. Оборудование производственное. Общие требования безопасности к рабочим местам».

27. СанПиН 2.2.3670-20 «Санитарно-эпидемиологические требования к условиям труда».

28. СП 52.13330.2011 «Естественное и искусственное освещение»

29. СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».

30. «ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты».

31. СП 12.13130.2009 «Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности».

Список публикаций

1. Kuligin S.M. Solving a quadratic programming problem based on the Gauss-Jordan transform / S.M. Kuligin // Recent Scientific Investigation: сб. ст. по материалам XXIInternationalMultidisciplinaryConferenceRecentScientificInvestigation Международной научно-практической конференции «Recent Scientific Investigation». – № 6(21). – М., Изд. «Интернаука», 2021.

Приложение А. Листинг программы

Файл **main.py**:

```
import numpy as np
import cv2 as cv
import utils
from table import Table
from PIL import Image
import xlsxwriter
import sys
from pdf2image import convert_from_path

# Загрузка файла
if len(sys.argv) < 2:
    print("Usage: python main.py <img_path>")
    sys.exit(1)

path = sys.argv[1]
if not path.endswith(".pdf") and not path.endswith(".jpg"):
    print("Must use a pdf or a jpg image to run the program.")
    sys.exit(1)

if path.endswith(".pdf"):
    ext_img = convert_from_path(path)[0]
else:
    ext_img = Image.open(path)

ext_img.save("data/target.png", "PNG")
image = cv.imread("data/target.png")
```

```
    NUM_CHANNELS = 3
    if len(image.shape) == NUM_CHANNELS:
        grayscale = cv.cvtColor(image, cv.COLOR_BGR2GRAY)

    # Фильтрация файла
    MAX_THRESHOLD_VALUE = 255
    BLOCK_SIZE = 15
    THRESHOLD_CONSTANT = 0

    filtered = cv.adaptiveThreshold(~grayscale, MAX_THRESHOLD_VALUE,
cv.ADAPTIVE_THRESH_MEAN_C, cv.THRESH_BINARY, BLOCK_SIZE,
THRESHOLD_CONSTANT)

    # Изоляция линий
    SCALE = 15

    horizontal = filtered.copy()
    vertical = filtered.copy()

    horizontal_size = int(horizontal.shape[1] / SCALE)
    horizontal_structure = cv.getStructuringElement(cv.MORPH_RECT,
(horizontal_size, 1))
    utils.isolate_lines(horizontal, horizontal_structure)

    vertical_size = int(vertical.shape[0] / SCALE)
    vertical_structure = cv.getStructuringElement(cv.MORPH_RECT, (1,
vertical_size))
    utils.isolate_lines(vertical, vertical_structure)

    # Выделение таблиц
```

```
mask = horizontal + vertical
(contours, _) = cv.findContours(mask, cv.RETR_EXTERNAL, cv.CHAIN_APPROX_SIMPLE)

intersections = cv.bitwise_and(horizontal, vertical)

tables = []
for i in range(len(contours)):
   (rect, table_joints) = utils.verify_table(contours[i], intersections)
   if rect == None or table_joints == None:
      continue

   table = Table(rect[0], rect[1], rect[2], rect[3])

   joint_coords = []

   for i in range(len(table_joints)):
      joint_coords.append(table_joints[i][0][0])
   joint_coords = np.asarray(joint_coords)
   sorted_indices = np.lexsort((joint_coords[:, 0], joint_coords[:, 1]))
   joint_coords = joint_coords[sorted_indices]

   table.set_joints(joint_coords)

   tables.append(table)

# Распознавание и запись
out = "bin/"
table_name = "table.jpg"
psm = 6
```

```
        oem = 3
        mult = 3

        utils.mkdir(out)
        utils.mkdir("bin/table/")

        utils.mkdir("excel/")
        workbook = xlsxwriter.Workbook('excel/tables.xlsx')

        for table in tables:
            worksheet = workbook.add_worksheet()
            table_entries = table.get_table_entries()
            table_roi = image[table.y:table.y + table.h, table.x:table.x + table.w]
            table_roi = cv.resize(table_roi, (table.w * mult, table.h * mult))

            cv.imwrite(out + table_name, table_roi)

            num_img = 0
            for i in range(len(table_entries)):
                row = table_entries[i]
                for j in range(len(row)):
                    entry = row[j]
                    entry_roi = table_roi[entry[1] * mult: (entry[1] + entry[3]) * mult,
entry[0] * mult:(entry[0] + entry[2]) * mult]
                    fname = out + "table/cell" + str(num_img) + ".jpg"
                    cv.imwrite(fname, entry_roi)
                    fname = utils.run_textcleaner(fname, num_img)
                    text = utils.run_tesseract(fname, num_img, psm, oem)
                    num_img += 1
                    worksheet.write(i, j, text)
```

```
workbook.close()
```

файл **utils.py**:

```
import cv2 as cv
import pytesseract as tess
from PIL import Image
import subprocess as s
import os

def isolate_lines(src, structuring_element):
    cv.erode(src, structuring_element, src, (-1, -1))
    cv.dilate(src, structuring_element, src, (-1, -1))

MIN_TABLE_AREA = 50
EPSILON = 3
def verify_table(contour, intersections):
  area = cv.contourArea(contour)

  if (area < MIN_TABLE_AREA):
    return (None, None)

  curve = cv.approxPolyDP(contour, EPSILON, True)

  rect = cv.boundingRect(curve)

  possible_table_region = intersections[rect[1]:rect[1] + rect[3], rect[0]:rect[0] + rect[2]]
  (possible_table_joints, _) = cv.findContours(possible_table_region, cv.RETR_CCOMP, cv.CHAIN_APPROX_SIMPLE)
```

```
        if len(possible_table_joints) < 5:
            return (None, None)

        return rect, possible_table_joints

    def mkdir(path):
        if not os.path.exists(path):
            os.makedirs(path)

    def showImg(name, matrix, durationMillis = 0):
        cv.imshow(name, matrix)
        cv.waitKey(durationMillis)

    def run_textcleaner(filename, img_id):
        mkdir("bin/cleaned/")

        cleaned_file = "bin/cleaned/cleaned" + str(img_id) + ".jpg"
        s.call(["./textcleaner", "-g", "-e", "none", "-f", str(10), "-o", str(5),
filename, cleaned_file])

        return cleaned_file


    def run_tesseract(filename, img_id, psm, oem):
        mkdir("bin/extracted/")

        image = Image.open(filename)
        language = 'eng'
        configuration = "--psm " + str(psm) + " --oem " + str(oem)
```

```
        text = tess.image_to_string(image, lang=language, config=configuration)
        if len(text.strip()) == 0:
            configuration += " -c tessedit_char_whitelist=0123456789"
            text = tess.image_to_string(image, lang=language, config=configuration)

        return text
```

файл **table.py**:

```
    class Table:
        def __init__(self, x, y, w, h):
            self.x = x
            self.y = y
            self.w = w
            self.h = h
            self.joints = None

        def __str__(self):
            return "(x: %d, y: %d, w: %d, h: %d)" % (self.x, self.x + self.w, self.y, self.y + self.h)

        def set_joints(self, joints):
            if self.joints != None:
                raise ValueError("Invalid setting of table joints array.")

            self.joints = []
            row_y = joints[0][1]
            row = []
            for i in range(len(joints)):
```

```
            if i == len(joints) - 1:
                row.append(joints[i])
                self.joints.append(row)
                break

            row.append(joints[i])

            if joints[i + 1][1] != row_y:
                self.joints.append(row)
                row_y = joints[i + 1][1]
                row = []

    def print_joints(self):
        if self.joints == None:
            print("Joint coordinates not found.")
            return

        print("[")
        for row in self.joints:
            print("\t" + str(row))
        print("]")

    def get_table_entries(self):
        if self.joints == None:
            print("Joint coordinates not found.")
            return

        entry_coords = []
        for i in range(0, len(self.joints) - 1):
```

```
            entry_coords.append(self.get_entry_bounds_in_row(self.joints[i],
self.joints[i + 1]))

        return entry_coords

    def get_entry_bounds_in_row(self, joints_A, joints_B):
        row_entries = []

        if len(joints_A) <= len(joints_B):
            defining_bounds = joints_A
            helper_bounds = joints_B
        else:
            defining_bounds = joints_B
            helper_bounds = joints_A

        for i in range(0, len(defining_bounds) - 1):
            x = defining_bounds[i][0]
            y = defining_bounds[i][1]
            w = defining_bounds[i + 1][0] - x
            h = helper_bounds[0][1] - y

            if h < 0:
                h = -h
                y = y - h

            row_entries.append([x, y, w, h])

        return row_entries
```

Приложение Б. (Справочное)

Neural Network Recognition of Financial Reporting Data of Russian Companies

Студент

Группа	ФИО	Подпись	Дата
0ВМ92	Кулигин Сергей Михайлович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Крицкий Олег Леонидович	к.ф-м.н.		

Консультант-лингвист отделения иностранных языков ШБИП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Утятина Янина Викторовна			

1. The practice

1.1 The Choosing a Programming Environment

One of the important aspects of qualitative model realization is the optimal software environment choice. Currently, there are a large number of programming languages and development environments.

As part of the master's dissertation, it is necessary to realize recognition of tables from an image and write the information to a file. Programming will be carried out using the Python language. Python is a general-purpose programming language that became very popular very quickly, mainly due to simplicity and readability of its code.

Compared to languages like C/C++, Python is slower. However, Python can be easily extended with C/C++, allowing to write resource-intensive C/C++ code and create Python wrappers that can be used like Python modules [7]. This has two advantages: first, code is running as fast as C/C++ source code, since it is actual C++ code running in background, and second, Python code is easier than C/C++.

OpenCV uses the Numpy library, which is a highly optimized library for numeric operations with MATLAB-style syntax. All OpenCV array structures are converted to and from Numpy arrays. It also makes it easier to integrate with other libraries like SciPy and Matplotlib which use Numpy.

1.2 Recognition Realization

To solve the problem, the OpenCV image processing library will mainly be used, as well as the Tesseract OCR text recognition library. In turn, Tesseract OCR uses neural networks to search and recognize text in an image and includes trained language models and different types of recognition [9].

The recognition realization of the necessary objects, in this case, tables, and the recording information from tables into a file consists of several stages, such as:

- Loading images;
- Image filtering;
- Lines isolation;
- Tables selection;
- Writing to a file.

Before starting the program, the path to the file to be processed will be provided.

1.2.1 Loading and Filtering Image

Speaking about downloading a file, we restrict ourselves to *pdf* and *jpg* file formats, since these formats are main ones when somebody is working with the program. If the file format doesn't fit the message like "The pdf or jpg format must be used." will appear and the program will be closed. The file is download using the cv.imread method. The file to be loaded is shown in Figure 2.1.

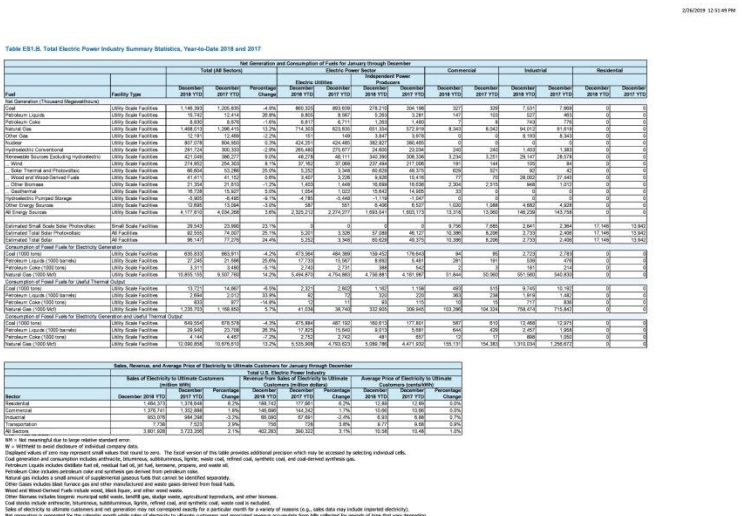

Figure 2.1 – Input data.

After downloading the file, an its copy in png format with name "target" will be created and then it will be used for work. This is nessesary to exclude possibility of losing the original file by editing it during the program execution.

The loaded image in the program will be represented as an mxn matrix

$$\begin{pmatrix} 1 & \dots & n \\ \dots & \dots & \dots \\ m & \dots & mn \end{pmatrix}, \quad (6)$$

where m is number of vertical pixels; n is number of pixels horizontally. Each individual element of the matrix is represented as a number from 0 to 255, which is colorness value of the pixel.

Basically, all images use three color channels: red, green and blue. Each image in the program will be encoded with three such matrices (for each of the colorness channels), therefore, for successful realiztion of image processing, it is necessary to convert the original image to grayscale. After that, there will be only one colorness parameter – brightness, which can take values from 0 to 255.

Filtering image means that we will get a different image with only white and black pixels. An input image can be colored, that is, it can has three color channels: red, green, blue, so it is necessary to convert it to grayscale using the cv.cvtColor method for further analysis.

White or black color of pixels is determined using a pixel brightness threshold. Brightness value of each pixel can range from 0 to 255, where 0 is black and 255 is white. Since OpenCV works with white objects on black background, we need to make pixels of the objects, which are supposed to contain the final result of the program, white, and others – black [10].

$$e_{ij} = \begin{cases} 255, e_{ij} = tv \\ 0, e_{ij} < tv \end{cases}, \quad (7)$$

Where e_{ij} – the element i row and j column;

tv – threshold value.

For our purposes, the threshold value is set 255, since we will check all getting contours. This means that for all pixels with brightness value equal 255, brightness value will be set 0, and for all other pixels – equal 255. The filtering is performed using the cv.adaptiveThreshold method, its result can be seen in Figure 2.2.

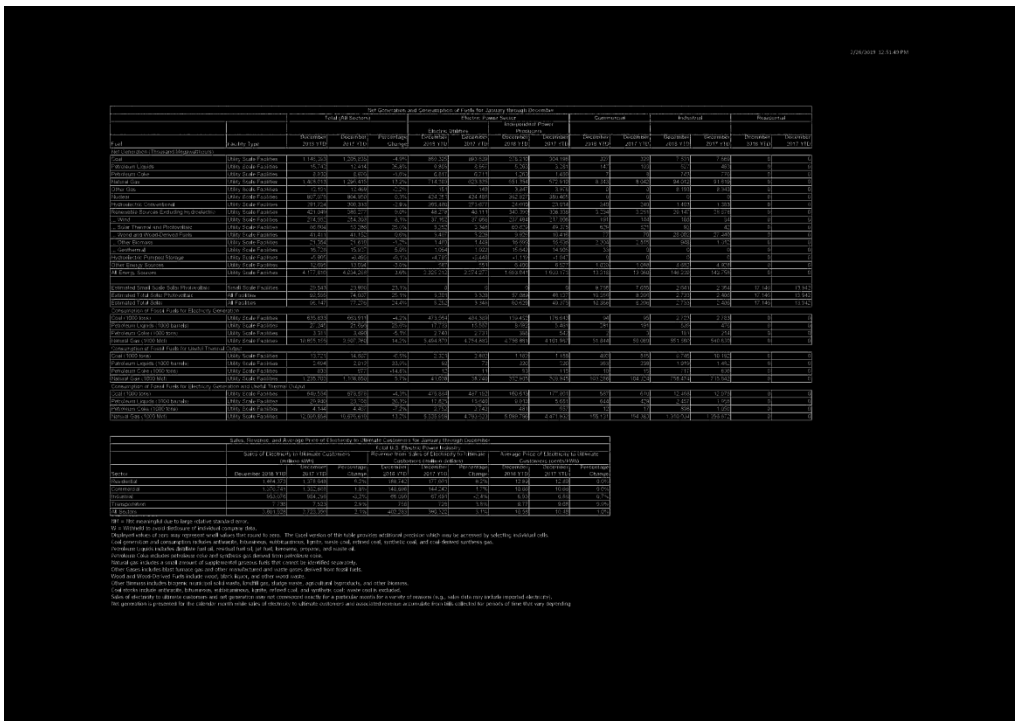

Figure 2.2 – Image filtering result.

As result, we obtain binary image. It is nessesary to make it possible to isolate lines.

Line isolation refers to a morphological transformation based on shape of an object in image [10]. Such transformation is carried out due to inconsistency of objects in the image with their expected shape, i.e. there are pixels that break structural integrality of the object's shape, for example, as shown in Figure 2.3.

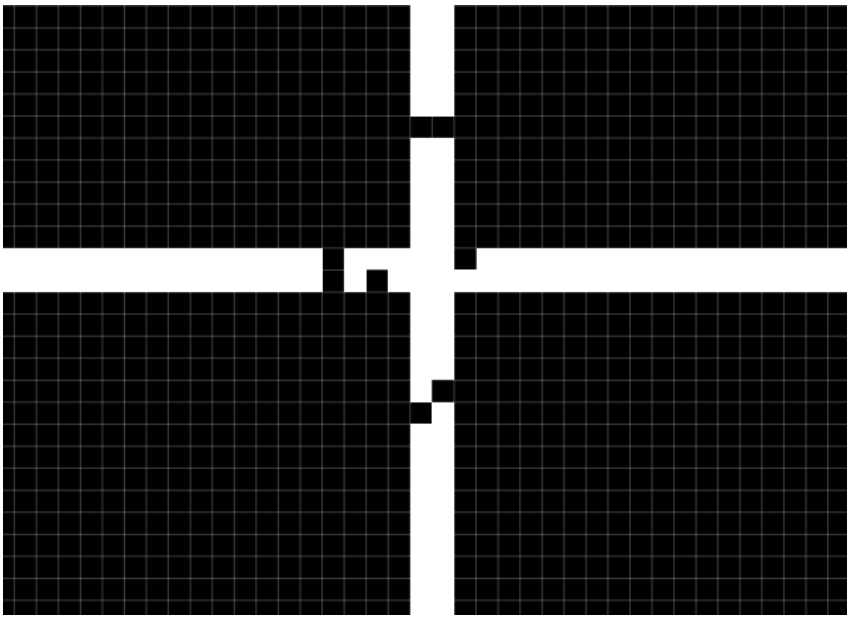

Figure 2.3 – Example of breaking structural integrality

Some pixels break rectangular shape of the object. To fix this problem, it is necessary to set color of the neighboring pixels to white, i.e. $e_{ij} = 255$. For visibility, an example of result of the morphological transformation is shown in Figure 2.4.

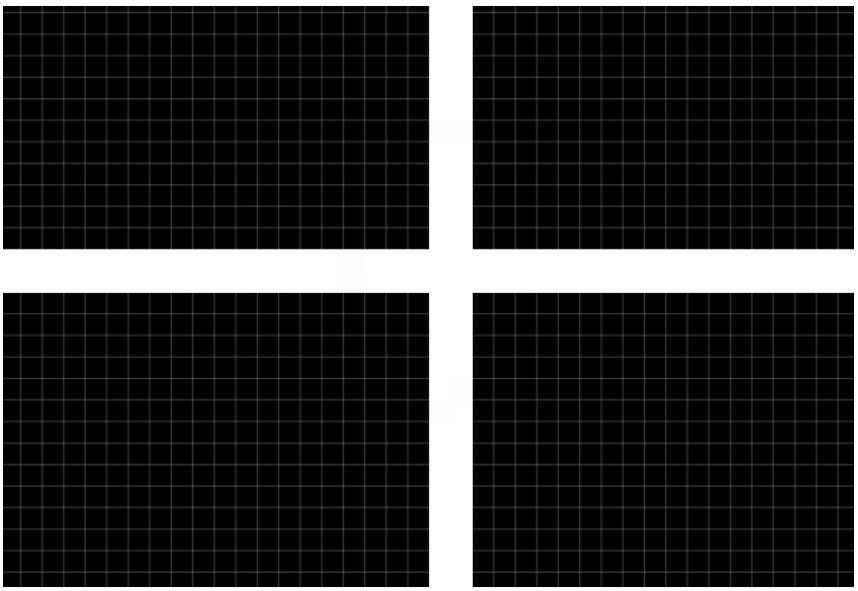

Figure 2.4 – Example of result of the morphological transformation.

The transformation receives two parameters as input – original image and structural element that determines nature of the transformation. In this case, it is necessary to take rectangular shape of structural element, since lines in the image should have such shape.

For transformation of image cv.getStructuringElement function will be used twice: for horizontal and vertical lines. Then we will find intersection of the two obtained images. Parameter cv.MORPH_RECT is passed to function input. It means that structuring element will be rectangular.

1.2.2 Tables Selection

Before selecting tables, we need to create a mask as sum of horizontal and vertical lines, find all its contours using cv.findContours function and find intersection of horizontal and vertical lines [14]. This is nessesary to check that the selected contours are indeed tables. This check includes the following:

- Checking for minimum possible number of pixels in contour area – 50;
- Contour approximation using Ramer-Douglas-Pecker algorithm;
- Finding the bounding rectangle of approximation points set;
- Finding number of connection points in each region of intersection;
- Checking for minimum possible number of connection points – 5.

If contour passes the check, a new instance of table with the coordinates of the bounding rectangle of the approximation points set will be created for it in the program code. Then, for this instance, the previously obtained coordinates of connection points, sorted from the last to the first, are specified.

1.2.3 Text Recognition and Writing to File

Tesseract uses language models and dictionaries to recognize text in specific language. Language model contains values of the parameters of the neural network model and other training data [15]. Before using model into practice, first we should train it using a different examples of symbols, both printed and handwritten. There is *EMNIST* database with large number of examples of printed and handwritten symbols. This database was created specifically for training models for text recognition.

Before training, all training images need to convert to *tif* format and combine into multi-page file. To complete training, we need to run the following commands in command line:

tesseract teslang.font.exp.tif teslang.font.exp nobatch box.train

unicharset_extractor teslang.font.exp.box

echo font 0 0 0 0 0>font_properties

mftraining -F font_properties -U teslang.unicharset -O teslang.unicharset teslang.font.exp.tr

cntraining teslang.font.exp.tr

combine_tessdata teslang

Thus, we will create new data model in Tesseract with name *teslang.traineddata*, and this model will be used for text recognition. To use the model, we will use predefined function run_tesseract in separate file:

```
def run_tesseract(filename, img_id, psm, oem):
    mkdir("bin/extracted/")
    image = Image.open(filename)
    language = 'teslang'
    configuration = "--psm " + str(psm) + " --oem " + str(oem)
    text = tess.image_to_string(image, lang=language, config=configuration)
    if len(text.strip()) == 0:
        configuration += " -c tessedit_char_whitelist=0123456789"
        text = tess.image_to_string(image, lang=language, config=configuration)
    return text
```

The recognition results are shown in Figure 2.5.

		Net Generation and Consumption of Fuels for December		
		Total (All Sectors)	Electric Power Sector	
			Electric Utilities	Independent Power Producers
Fuel	Facility Type	December December Percentage 2018 2017 Change	December December 2018 2017	December December 2018 2017
Net Generation (Thousand Megawatthours)				
Coal	Utility Scale Facilities	96825	106546	-9,10%
Petroleum Liquids	Utility Scale Facilities	930	1982	-53,10%
Petroleum Coke	Utility Scale Facilities	807	737	9,50%
Natural Gas	Utility Scale Facilities	106978	111373	-3,90%
Other Gas	Utility Scale Facilities	998	1096	-9,00%
Nuclear	Utility Scale Facilities	71657	73700	-2,80%
Hydroelectric Conventional	Utility Scale Facilities	23728	22377	6,00%
Renewable Sources Excluding Hydroelectric	Utility Scale Facilities	34787	35151	-1,00%
... Wind	Utility Scale Facilities	24825	24575	1,00%
... Solar Thermal and Photovoltaic	Utility Scale Facilities	3188	3389	-5,90%
... Wood and Wood-Derived Fuels	Utility Scale Facilities	3414	3738	-8,70%
... Other Biomass	Utility Scale Facilities	1825	1877	-2,80%
... Geothermal	Utility Scale Facilities	1535	1571	-2,30%
Hydroelectric Pumped Storage	Utility Scale Facilities	-522	-656	-20,40%
Other Energy Sources	Utility Scale Facilities	1147	1146	0,10%
All Energy Sources	Utility Scale Facilities	337334	353452	-4,60%
Estimated Small Scale Solar Photovoltaic	Small Scale Facilities	1774	1472	20,50%
Estimated Total Solar Photovoltaic	All Facilities	4870	4739	2,80%
Estimated Total Solar	All Facilities	4962	4861	2,10%
Consumption of Fossil Fuels for Electricity Generation				

Figure 2.5 – Recognition results.

In Figure 2.5, we can see that the recognition result is quite accurate. Although some lines contain wrong recognized symbol "...", the most important criterion – possibility of more convenient work with data, has been completed.

1.3 Accuracy Assessment

To understand how accurate result which the program gives, we will compare result with simple manual data entry.

To assess the recognition accuracy, we calculate the total number of characters n_{gen} and the number of errors n_{er}. Then recognition accuracy is calculated using formula

$$A = \frac{n_{gen} - n_{er}}{n_{gen}} * 100\%. \quad (8)$$

Recognition time for one page is calculated using formula

$$t_p = \frac{t_{gen}}{N}, \quad (9)$$

where N – number of recognized pages; t_{gen} – recognition time of all pages.

To carry out accuracy research, we used the monthly financial report on electricity of PJSC Gazprom for december 2018. We made several recognition attempts with gradual increase number of pages. The results of accuracy assessment are presented in Table 1.

Table 1. Quantitative characteristics of recognition quality by the program

Number of recognized pages, N	Total number of characters n_{gen}	Number of errors n_{er}	Recognition time for one page t_p, c	Recognition accuracy A, %
5	13 858	41	2.694	99.7
10	29 322	104	2.994	99.6
15	42 117	142	3,154	99.7
20	51 656	182	3.403	99.6
25	62 050	213	3,502	99.7

Based on the obtained indicators, we can draw a short conclusion. The accuracy of symbols recognition in tables is on average 99.7%, which is a very good result, given that the selection and processing of information during recognition is very capacious and time-consuming process. This is emphasized that the recognition time for one page increases with the number of recognized pages.

Such data will allow us to analyze and plan time costs for large capacity of recognizable material.

References

1. Klette R. Computer vision. Theory and algorithms, – Moscow: DMK Press, 2019. – 506 p.
2. Shakiryanov E. D. Computer vision in Python. The first steps, – Moscow: Laboratoria znanii, 2021. – 160 p.
3. Solem Y. E. Computer vision programming in Python, – Moscow: DMK Press, 2016. – 314 p.
4. Charu A. Neural networks and deep learning. Training course, – Moscow: Vilyams, 2020. – 752 p.
5. Understanding LSTM Networks. Available at: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed 16.04.2021).
6. Douglas-Pecker algorithm. Available at: https://habr.com/ru/post/448618/ (accessed 23.03.2021).
7. Mueller A. Introduction to machine learning with Python. Data Scientist's Guide, – Moscow: Vilyams, 2017. – 480 p.
8. Lutz M. Learning Python, – Moscow: Symbol-Plus, 2011. – 1280 p.
9. Francya C. Deep Learning in Python, – St. Petersburg.: Piter, 2018. – 400 p.
10. Woods R., Gonzalez R. Digital Image Processing – Moscow: Tehnosfera, 2012. – 1104 p.
11. Rashid T. Create a neural – Moscow: Vilyams, 2018. – 272 p.
12. Neural networks. Part. Available at: https://habr.com/ru/post/312450/ (accessed 04.04.2021)
13. Writing your own neural network: a step-by-step. Available at: https://proglib.io/p/neural-nets-guide/ (accessed 30.03.2021).
14. Recognition of images using. Available at: http://recog.ru/wp-content/uploads/2020/05/opencvkruchinin.pdf (accessed 21.02.2021)
15. Tesseract User Manual. Available at: https://github.com/tesseract-ocr/tessdoc#tesseract-user-manual (accessed 26.04.2021).