МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ
ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
Национальный исследовательский ядерный университет «МИФИ»
Институт Лазерных и Плазменных Технологий
Кафедра № 97 «Суперкомпьютерное моделирование
инженерно-физических процессов»

ОТЧЁТ
Учебная практика (научно-исследовательская работа) на тему:

Подготовка базы свойств материалов на основе открытых источников

Работу выполнил:
А. Е. Диков
Группа: Б20-221
Научный руководитель:
Д. П. Макаревич

Москва
2023

Содержание

1. Введение

1.1. Актуальность проблемы

В современном информационном обществе существует огромное количество данных, которые хранятся в различных форматах, включая оцифрованные изображения документов. Вместе с тем, большое количество информации представлено в виде таблиц, которые являются основным средством структурирования и организации данных. Однако, извлечение данных из изображений таблиц является нетривиальной задачей, которая требует автоматизации и использования специализированных технологий.

Проблема извлечения данных из изображений таблиц имеет высокую актуальность в различных сферах деятельности, таких как научные исследования, финансовый анализ, медицина, юридические и бухгалтерские отчеты и многие другие. Сфокусируем внимание на применении в научной сфере, а именно, для сбора данных о параметрах материалов, используемых в матемаческом моделировании. Автоматическое распознавание и извлечение данных из изображений таблиц позволяет значительно ускорить и упростить процесс обработки информации.

1.2. Трудности реализации задачи

Задача извлечения данных из изображений таблиц является сложной и требует решения нескольких проблем. Во-первых, требуется точное обнаружение и сегментация таблицы на изображении. Это включает в себя предобработку изображений, выделение контуров, определение границ и разделение на ячейки.

Во-вторых, необходимо правильно распознать текст и числа внутри каждой ячейки таблицы. Распознавание текста визуально представленного на изображении является сложной задачей из-за различных шрифтов, размеров, стилей и размещения текста в ячейках таблицы.

В-третьих, требуется структурировать распознанные данные в виде таблицы с соответствующими заголовками столбцов и строками. Иными словами, сохранить общий вид исходной таблицы.

1.3. Используемые технологии

Для решения задачи извлечения данных из изображений таблиц используются различные технологии и методы. Ниже приведены некоторые из них:

1.3.1. Обработка изображений с использованием OpenCV

OpenCV (Open Source Computer Vision Library) является библиотекой с открытым исходным кодом на языке C++, предназначенной для обработки изображений и компьютерного зрения. Она предоставляет мощные инструменты для обработки и анализа изображений, включая выделение контуров, сегментацию, фильтрацию и морфологические операции над изображениями.

1.3.2. Распознавание текста с помощью библиотеки Tesseract

Tesseract - это библиотека с открытым исходным кодом для распознавания текста на изображениях. Она использует алгоритмы машинного обучения для распознавания текста и может работать с различными языками. Tesseract обладает хорошей точностью распознавания и широкими возможностями настройки.

Ядро Tesseract было разработано в лаборатории Hewlett Packard в середине 1980-х и поддерживалось до середины 1990-х, а затем 10 лет никакой поддержки не было. С 2006 г. Google купил Tesseract и открыл исходные тексты под лицензией Apache 2.0 для продолжения разработки. В настоящий момент программа поддеживает кодировку UTF-8 и множество языков, включая русский.

1.3.3. Использование нейронных сетей

Нейронные сети, такие как Faster R-CNN, SSD, YOLO и RetinaNet, являются мощными инструментами для обнаружения объектов и сегментации на изображениях. Они используют алгоритмы глубокого обучения для автоматического обнаружения и распознавания объектов, включая таблицы и их компоненты. [пока не функционируют, из использовние в планах]

2. Постановка задачи

В качестве набора данных были предоставлены ГОСТ, технические условия и справочники в различных форматах - rtr, djvu, pdf. В дополнению к этому был сгенерирован еще один набор данных, имеющий меньшую вариативность наполнения, но имеющий метки - файлы в более удобном формате (был выбран csv) для оценки качества работы алгоритма (размеченные данные). Задача состояла в выделении таблицы из корпуса текста в виде изображений. Затем нужно было распознать таблицу с помощью алгоритмов компьютерного зрения так, чтобы сохранилась общая сруктура таблицы. Необходимо было создать универсальную и масштабируемую систему, точно работающую для всех входных файлов.

Облучаемая группа населения	Коэффициент риска злокачественных новообразований, x10-2 Зв-1	Коэффициент риска наследственных эффектов, x10-2 Зв-1	Сумма, x10-2 Зв-1
Все население	5,5	0,2	5,7
Взрослые	4,1	0,1	4,2

Рисунок 2.1. Пример таблицы (взят из НРБ 99/2009)

	C1	C2	C3	C4
1	Облучаемая группа населения	Коэффициент риска злокачественных новообразований, x10-2 Зв-1	Коэффициент риска наследственных эффектов, x10-2 Зв-1	Сумма, x10-2 Зв-1
2	Все население	5,5	0,2	5,7
3	Взрослые	4,1	0,1	4,2

Рисунок 2.2. Результат работы алгоритма

3. Алгоритмы, технологии и программный код

3.1. Обработка входных данных

Для решения задачи напишем класс на языке Python, полями которого будут подоваемый на вход файл, из которого нужно извлечь таблицы, и базы данных (pandas.dataFrame) в которых будет храниться результат работы.

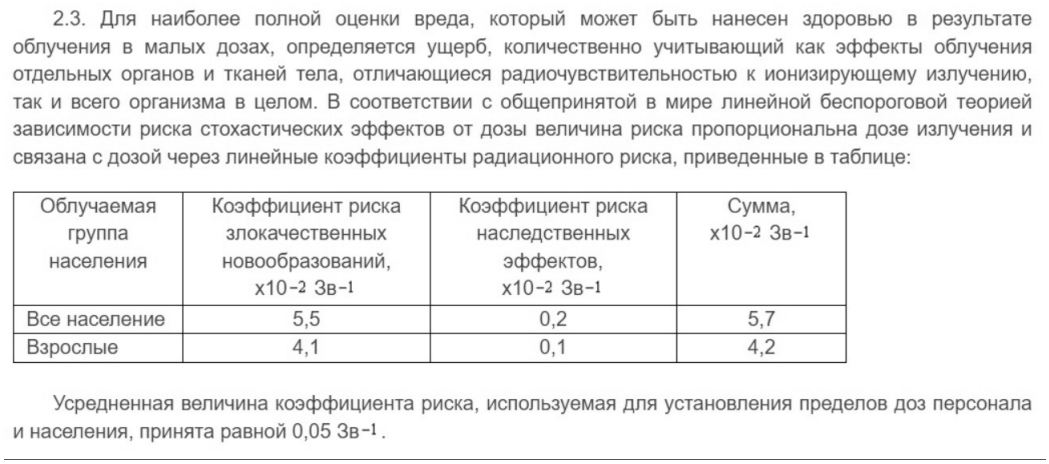
```
class TableExtraction:
    def __init__(self):
        self.input_data = []
        self.list_of_np_arrays = []
        self.threshold_images = []
        self.parts_boxes = []
        self.parts_image = []
        self.result = []

    def convert_file_to_array(self, file_path):...

    def binarization(self):...

    def split_pages(self):...

    def extract(self):...
```

Пока для упрощения будем считать что входной файл всегда имеет формат pdf.

```
def convert_file_to_array(self, file_path):
    file_extension = os.path.splitext(file_path)[1]
    if file_extension == '.pdf':
        doc = fitz.open(file_path)
        for n in range(doc.page_count):
            page = doc.load_page(n)
            pix = page.get_pixmap()
            image = np.frombuffer(pix.samples, dtype=np.uint8).reshape(pix.h,
                pix.w, pix.n)
            image = np.ascontiguousarray(image[..., [2, 1, 0]])
            self.list_of_np_arrays.append(image)
        doc.close()
        else:
            raise ValueError('Unsupported file format')
```

Отсканированные изображения страниц документов в общем случае могут быть цветными, что затормозит работу алгоритмов поиска контуров и распозования букв в дальнейшем, поэтому опустим информацию о цвете. Более того обросим и оттенки серого, тем самым бинаризовав изображение.

```
def binarization(self):
    for image in self.list_of_np_arrays:
        gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        _, threshold_image = cv2.threshold(gray_image, 0, 255,
            cv2.THRESH_BINARY)
            self.threshold_images.append(threshold_image)
```

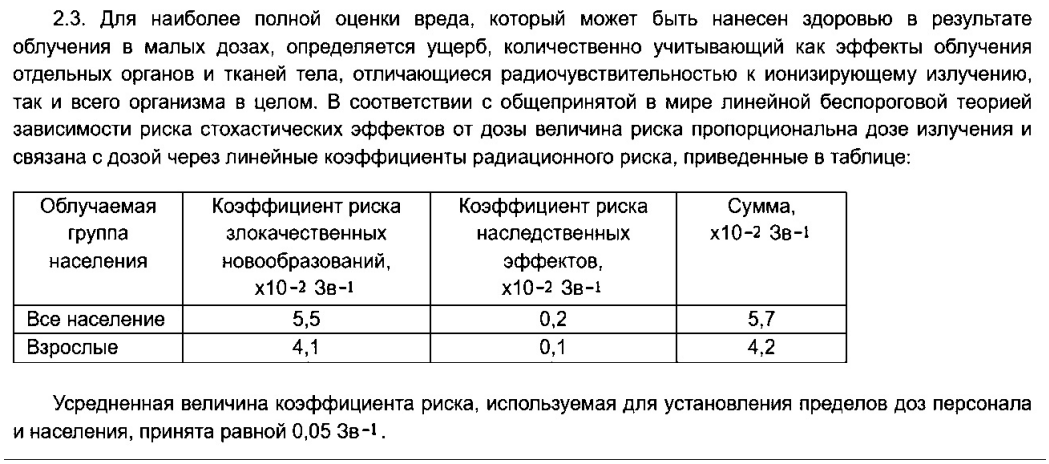


2.3. Для наиболее полной оценки вреда, который может быть нанесен здоровью в результате облучения в малых дозах, определяется ущерб, количественно учитывающий как эффекты облучения отдельных органов и тканей тела, отличающиеся радиочувствительностью к ионизирующему излучению, так и всего организма в целом. В соответствии с общепринятой в мире линейной беспороговой теорией зависимости риска стохастических эффектов от дозы величина риска пропорциональна дозе излучения и связана с дозой через линейные коэффициенты радиационного риска, приведенные в таблице:

Облучаемая группа населения	Коэффициент риска злокачественных новообразований, x10-2 Зв-1	Коэффициент риска наследственных эффектов, x10-2 Зв-1	Сумма, x10-2 Зв-1
Все население	5,5	0,2	5,7
Взрослые	4,1	0,1	4,2

Усредненная величина коэффициента риска, используемая для установления пределов доз персонала и населения, принята равной 0,05 Зв-1.

Рисунок 3.1. Черно-белое изображение



2.3. Для наиболее полной оценки вреда, который может быть нанесен здоровью в результате облучения в малых дозах, определяется ущерб, количественно учитывающий как эффекты облучения отдельных органов и тканей тела, отличающиеся радиочувствительностью к ионизирующему излучению, так и всего организма в целом. В соответствии с общепринятой в мире линейной беспороговой теорией зависимости риска стохастических эффектов от дозы величина риска пропорциональна дозе излучения и связана с дозой через линейные коэффициенты радиационного риска, приведенные в таблице:

Облучаемая группа населения	Коэффициент риска злокачественных новообразований, x10-2 Зв-1	Коэффициент риска наследственных эффектов, x10-2 Зв-1	Сумма, x10-2 Зв-1
Все население	5,5	0,2	5,7
Взрослые	4,1	0,1	4,2

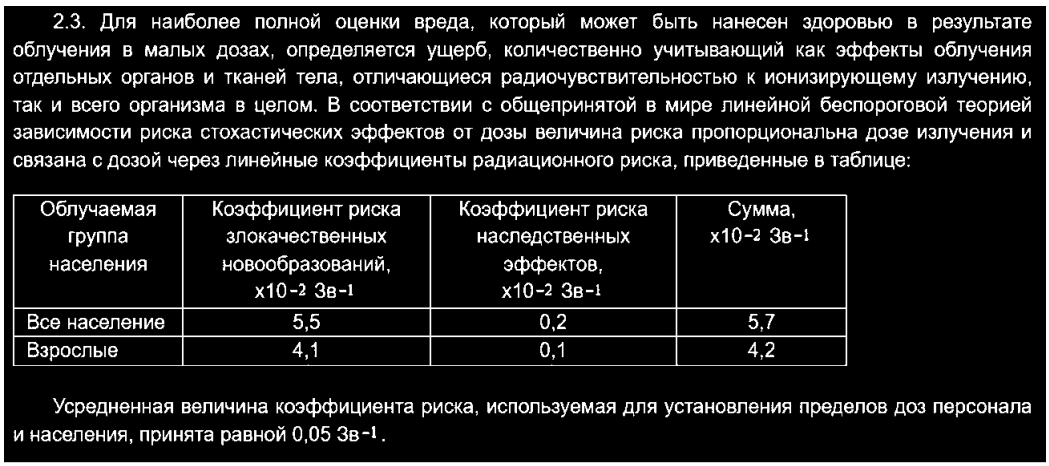
Усредненная величина коэффициента риска, используемая для установления пределов доз персонала и населения, принята равной 0,05 Зв-1.

Рисунок 3.2. Бинаризованное изображение

3.2. Поиск таблиц в документе

Теперь будем рассматривать документ как набор черно-белых изображений и все сказанное ниже будем применять для каждой страницы документа. Для нахождения таблицы на страницы сделаем копию изображения и применим к ней инфертированную бинаризацию и морфологическое сверточное спреобразование. Оно размоет границы, замкнет линии и уменьшит возможные шумы, возникающие в результате сканирования.

```
1 kernel = np.ones((5, 5), np.uint8)
2 morph_image = cv2.morphologyEx(threshold_image, cv2.MORPH_CLOSE, kernel)
```



2.3. Для наиболее полной оценки вреда, который может быть нанесен здоровью в результате облучения в малых дозах, определяется ущерб, количественно учитывающий как эффекты облучения отдельных органов и тканей тела, отличающиеся радиочувствительностью к ионизирующему излучению, так и всего организма в целом. В соответствии с общепринятой в мире линейной беспороговой теорией зависимости риска стохастических эффектов от дозы величина риска пропорциональна дозе излучения и связана с дозой через линейные коэффициенты радиационного риска, приведенные в таблице:

Облучаемая группа населения	Коэффициент риска злокачественных новообразований, x10-2 Зв-1	Коэффициент риска наследственных эффектов, x10-2 Зв-1	Сумма, x10-2 Зв-1
Все население	5,5	0,2	5,7
Взрослые	4,1	0,1	4,2

Усредненная величина коэффициента риска, используемая для установления пределов доз персонала и населения, принята равной 0,05 Зв-1.

Рисунок 3.3. Инвертированно бинаризованное изображение

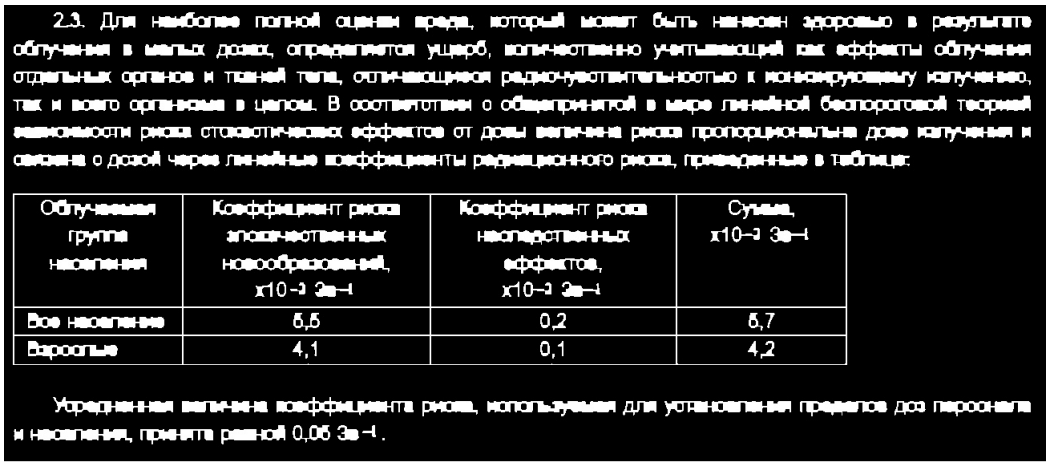


Рисунок 3.4. Изображение, после применения морфологического преобразования

Применим к предобработанному изображению метод findContours(). Внутри метода findContours() в OpenCV используется алгоритм обхода границы объектов, известный как алгоритм Сузуки и Абе. [ссылка]

Алгоритм работает следующим образом:

1. Имеется бинарное изображение, где объекты обозначаются белым цветом, а фон - черным цветом. Каждый пиксель бинарного изображения имеет значение 0 или 1.

2. Сканируем изображение и находим первую непосещенную точку на границе объекта (белый пиксель) и присваивает ей имя.

3. Следует по границе объекта, перемещаясь от текущей точки к следующей, пока не вернется в исходную точку. Каждая точка границы добавляется в список контура.

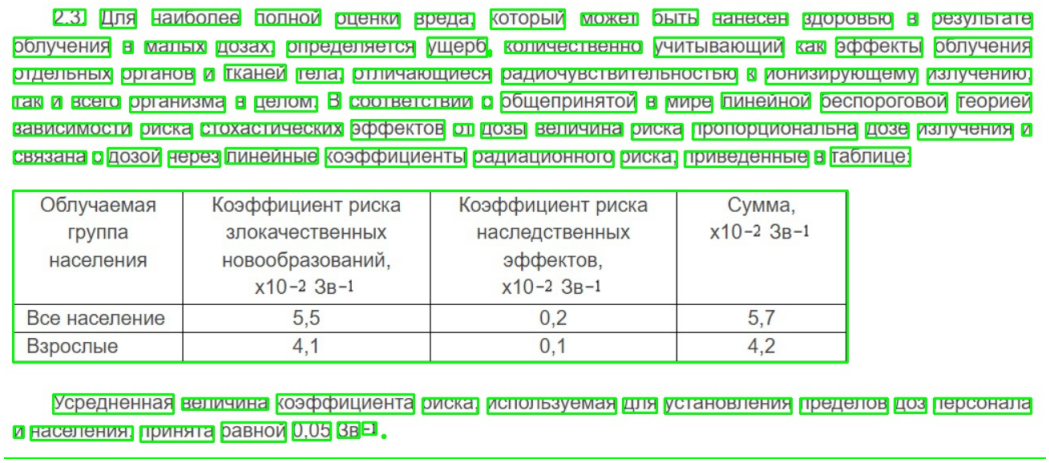
4. Маркируем посещенные точки на границе объекта, чтобы не посещать их повторно.

5. Повторение для оставшихся объектов: Процесс повторяется для остальных непосещенных объектов на изображении, пока не будут обнаружены все контуры.

Особенности алгоритма, связанные с восстановлением иерархии границ, при решении задачи не понадобятся, поэтому их опустим.

```
1 contours, hierarchy = cv2.findContours(morph_image, cv2.RETR_EXTERNAL,
    cv2.CHAIN_APPROX_SIMPLE)
```

Здесь режим `cv2.RETR_EXTERNAL` позволяет извлечь только внешние контуры, игнорируя внутренние, а `cv2.CHAIN_APPROX_SIMPLE` возвращает координаты ограничивающих эти контуры прямоугольников



2.3. Для наиболее полной оценки вреда, который может быть нанесен здоровью в результате облучения в малых дозах, определяется ущерб, количественно учитывающий как эффекты облучения отдельных органов и тканей тела, отличающиеся радиочувствительностью к ионизирующему излучению, так и всего организма в целом. В соответствии с общепринятой в мире линейной беспороговой теорией зависимости риска стохастических эффектов от дозы величина риска пропорциональна дозе излучения и связана с дозой через линейные коэффициенты радиационного риска, приведенные в таблице:

Облучаемая группа населения	Коэффициент риска злокачественных новообразований, x10-2 Зв-1	Коэффициент риска наследственных эффектов, x10-2 Зв-1	Сумма, x10-2 Зв-1
Все население	5,5	0,2	5,7
Взрослые	4,1	0,1	4,2

Усредненная величина коэффициента риска, используемая для установления пределов доз персонала и населения, принята равной 0,05 Зв-1.

Рисунок 3.5. Результат работы cv2.findContours

Далее надо провести фильтрацию контуров. Заметим что таблица занимает большее пространство чем каждое из слов и введем эмперический критерий по площади контура. Сохраним, как координаты предполагаемой таблицы, так и вырезанное со страницы ее изображение.

```
1 for contour in contours:
2     area = cv2.contourArea(contour)
3     if area > 3000:
4         x, y, w, h = cv2.boundingRect(contour)
5         self.parts_boxes.append((x, y, x + w, y + h))
6         x1, y1, x2, y2 = self.parts_boxes[-1]
7         self.parts_image.append(image[y1:y2, x1:x2])
```

3.3. Работа с таблицей

Далее будем работать в основном с библиотекой Tesseract, поэтому обозначим что внутри этой библиотеки есть множество методов машинного обучения и специализированных алгоритмов, применяемых для конкретных этапов обработки текста. После сегментации текста, алоритмом аналогичным описанному выше алгоритму Сузуки и Абе, изображения символов поступают в сверточную рекуренстую нейронную сеть для классификации. Сверточные преобразовании ползволяют получить инвариантность относительно перобразований поворота, сдвига и искажений, в то время как рекуррентность дает лучшие результаты исходя из того, что текст это последовательность - ряд - в котором предыдущие элементы влияют на последующие. Архитектура этой сети фиксированная, сеть пердобучена на большом наборе данных и дает хорошую точность.

4. Результаты

Запустить на датасете, сравнить, ввести метрику

5. Заключение

Что получилось и не получилось что можно улучшить что можно поменять