**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский

технический университет им. А.Н. Туполева-КАИ»

(КНИТУ-КАИ)

Институт компьютерных технологий и защиты информации

Кафедра систем автоматизированного проектирования

09.03.01. «Информатика и вычислительная техника»

Курсовая работа

по дисциплине: «Разработка систем автоматизированного проектирования»  
на тему: «Разработка подсистемы распознавания рисунка проводящего монтажа на послойный чертежах печатных плат»

Обучающийся 4314 \_\_\_\_\_\_\_\_\_\_\_\_ Бикмуллин А.Р

(номер группы) (подпись, дата) (Ф.И.О.)

Руководитель старший преподаватель Суздальцев И.В.

(должность) (Ф.И.О.)

Курсовая работа (проект) зачтена (зачтен) с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

Казань 2023

**СОДЕРЖАНИЕ**

Введение

Глава 1. Анализ проектной процедуры разработки подсистемы распознавания рисунка проводящего монтажа на послойный чертежах печатных плат.

1.1. Разработка функциональных моделей (IDEF0)

1.2. Разработка поведенческих моделей (IDEF3)

Глава 2. Разработка математического обеспечения

2.1. Содержательная постановка задачи

2.2. Математическая постановка задачи

2.3. Разработка модели для задачи распознавания рисунка проводящего монтажа на послойный чертежах печатных плат

2.4. Описание метода обучения нейронной сети.

2.5. Решение задачи на контрольном примере

Глава 3. Разработка информационного обеспечения

3.1 Разработка модели потока данных (DFD) проектной процедуры распознавания рисунка проводящего монтажа на послойный чертежах печатных плат

3.2. Разработка концептуальной модели базы данных

3.3. Разработка логической модели базы данных

3.4. Физическое проектирование базы данных

Заключение

Список источников

ПРИЛОЖЕНИЕ

# **ВВЕДЕНИЕ**

В современных технологиях электроники печатные платы играют важную роль в создании и производстве электронных устройств. Одним из важных этапов в процессе разработки печатных плат является проведение монтажа компонентов на поверхности платы. Правильное расположение и подключение компонентов является необходимым условием для достижения надежной работы электронных устройств. Однако, в процессе разработки и производства печатных плат возникает множество сложностей, связанных с распознаванием и интерпретацией сложных чертежей печатных плат. Проводящие монтаж на послойных чертежах печатных плат содержит множество элементов и соединений, что делает визуальный анализ таких чертежей трудоемким и подверженным ошибкам. В связи с этим, разработка подсистемы распознавания рисунка проводящего монтажа на послойных чертежах печатных плат становится актуальной задачей. Данная подсистема будет позволять автоматически анализировать и интерпретировать чертежи печатных плат, определять расположение компонентов и их соединения.  
 Целью данной курсовой работы является разработка и реализация подсистемы распознавания рисунка проводящего монтажа на послойных чертежах печатных плат. Для достижения этой цели необходимо решить следующие задачи:

1. Разработка функциональных моделей (IDEF0)
2. Разработка поведенческих моделей (IDEF3)
3. Содержательная постановка задачи трассировки
4. Математическая постановка задачи трассировки
5. Выбор и описание алгоритма решения задачи
6. Разработка модели для задачи распознавания рисунка проводящего монтажа на послойный чертежах печатных плат
7. Описание метода обучения нейронной сети.
8. Решение задачи на контрольном примере
9. Разработка модели потока данных (DFD) проектной процедуры распознавания рисунка проводящего монтажа на послойный чертежах печатных плат
10. Разработка концептуальной модели базы данных
11. Разработка логической модели базы данных
12. Физическое проектирование базы данных

**Глава 1.** **Анализ проектной процедуры разработки подсистемы распознавания рисунка проводящего монтажа на послойный чертежах печатных плат.**

**1.1. Разработка функциональных моделей (IDEF0).**

В качестве входных потоков информации представлены:

* *Справочная информация* – это стандарты и спецификации, техническая документация, тестовые данные. Используется *ГОСТ 2.726-2014 "Единая система конструкторской документации (ЕСКД). Требования к оформлению документов в электронной форме", а также ГОСТ 2.103-2013 "Единая система конструкторской документации (ЕСКД). Общие требования к чертежам".*
* *Изображение послойных чертежей печатных плат* – это графическое изображение структуры печатной платы, состоящей из нескольких слоев, на которых нанесены элементы монтажа и проводники, соединяющие эти элементы. Каждый слой представлен на чертеже отдельно, что позволяет увидеть, какие элементы расположены на каждом слое и как проводники проходят через слои, соединяя элементы монтажа. На изображении послойного чертежа печатной платы могут быть изображены различные элементы монтажа, такие как микросхемы, резисторы, конденсаторы, разъемы и др., а также проводники, соединяющие эти элементы между собой и с другими узлами на плате. Используется *ГОСТ 2.720-2011 ЕСКД. Общие требования к выполнению чертежей*

Выходные параметры данных будут следующими:

* Рисунок проводящего монтажа в формате “Gerber” — это документ, который включает в себя графическую информацию о всех компонентах проводящего монтажа. Формат “Gerber” (Gerber format) является стандартным форматом файлов для представления информации о печатных платах, включая проводящий монтаж. В формате “Gerber” каждый слой печатной платы представляется отдельным файлом.  
    
  Каждый файл “Gerber” содержит информацию о различных элементах проводящего монтажа, таких как печатные проводники, отверстия, компоненты и т. д. Файлы “Gerber” имеют расширения ".gbr" или ".ger".

В качестве регламентирующей документации используются такие данные как:

* *ГОСТ 2.721-74. ЕСКД Обозначения условные графические в схемах. Обозначения общего применения.* Настоящий стандарт устанавливает условные графические обозначения общего применения на схемах, выполняемых вручную или автоматизированным способом, изделий всех отраслей промышленности и строительства.
* *ГОСТ 2.105-95 ЕСКД Общие требования к текстовым документам.* Настоящий стандарт устанавливает общие требования к выполнению текстовых документов изделия машиностроения, приборостроения и строительства.
* *ГОСТ 2.710-81 ЕСКД Обозначения буквенно-цифровые в электрических схемах.* Настоящий стандарт распространяется на электрические схемы, а также на конструкторские документы, содержащие сведения об элементах, устройствах и функциональных группах электрических схем, выполняемых вручную и автоматизированным способом во всех отраслях промышленности, и устанавливает типы условных буквенно-цифровых обозначений элементов, устройств и функциональных групп, а также правила их построения.
* *2.702-2011 ЕСКД Правила выполнения принципиальных электрических схем.* Настоящий стандарт распространяется на электрические схемы изделий всех отраслей промышленности, а также электрические схемы энергетических сооружений и устанавливает правила их выполнения.

В качестве потоков механизмов будут использоваться такие средства как:

* *ЭВМ* – это компьютерное устройство, предназначенное для автоматизации обработки информации и выполнения различных вычислительных операций.
* *Инженеры-проектировщики* – это специалисты, занимающиеся проектированием и разработкой различных систем, конструкций и устройств. Они обладают знаниями в области технического рисования, материаловедения, механики, электроники, автоматизации и других инженерных наук. Диаграмма IDEF0 (А – 0) изображена на рис. 1.

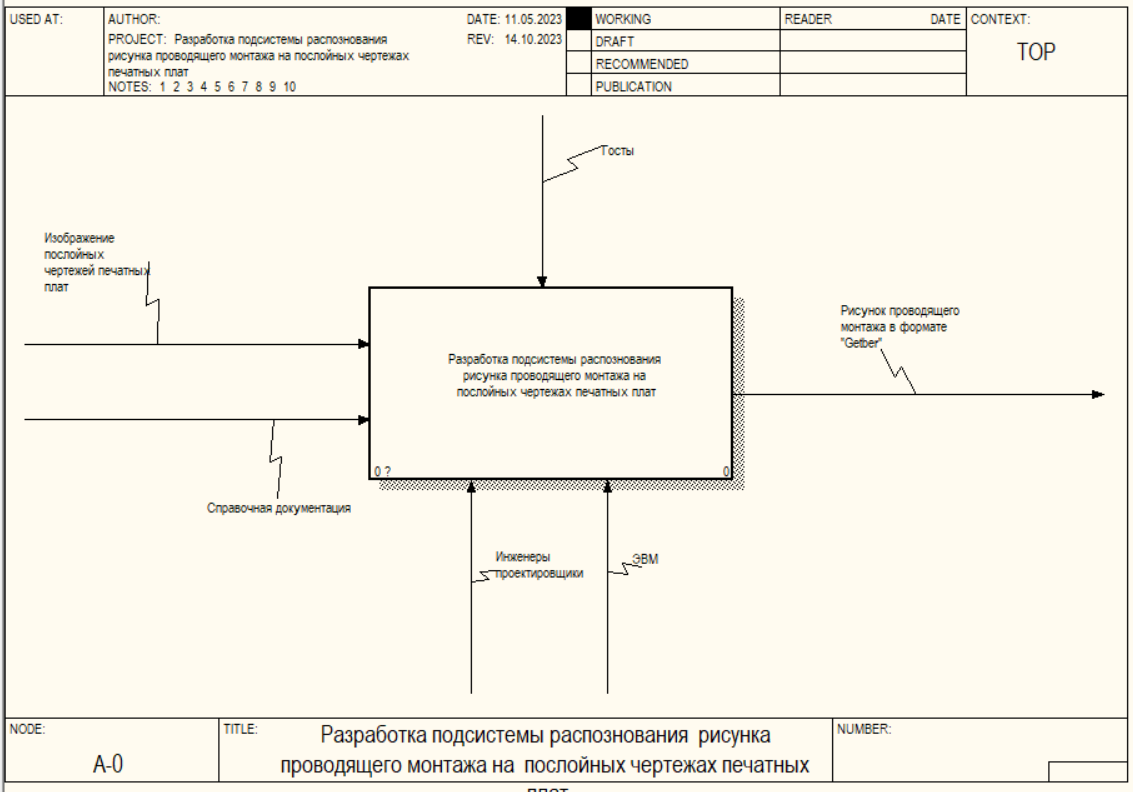


Рис.1 Контекстная диаграмма A-0

## **Описание IDEF0 A0 диаграммы**

1. Блок «Процедуры подготовки исходных данных, верификации и утверждения проектной документации». Этап включает в себя ряд действий и процессов, которые выполняются для обеспечения правильности, соответствия требованиям и готовности проектной документации перед ее передачей на следующие этапы разработки или производства. Данный этап включает в себя задачи: подготовка исходных данных, разработка проектной документации, верификация проектной документации, утверждение проектной документации.

В качестве входных потоков информации представлены:

* *Изображение послойных чертежей печатных плат;*
* *Справочная документация;*
* *Перечень координат межсоединительных связей*
* *Результат распознавания рисунка;*

В качестве выходных потоков информации представлены:

* *Рисунок проводящего монтажа в формате “Gerber”;*
* *Улучшенное визуальное изображение печатной платы* (процедура улучшения качества изображения включает в себя: изменение параметров элементов, отображение слоев, восстановление потерянных пикселей, иногда увеличение качества изображения (увеличение количества пикселей));

В качестве регламентирующей документации представлены:

* *ГОСТы;*

В качестве потоков механизмов будут использоваться такие средства как:

* *Инженеры-проектировщики*;
* *ЭВМ;*

1. Блок «Процедура обнаружения межсоединительных связей и извлечения их свойств с использованием нейронной сети». На данном этапе происходит процесс распознавания элементов по определенному алгоритму, выполняемому ЭВМ.

В качестве входных потоков информации представлены:

* *Улучшенное визуальное изображение печатной платы;*

В качестве выходных потоков информации представлены:

* *Перечень координат межсоединительных связей (это результат распознавания межсоединительных связей на рисунке и их представления в формате (x,y))*
* *Результат распознавания рисунка (это рисунок печатной платы на котором присутствуют распознанные проводники, обозначения и метки, слои и структура, ошибки и несоответствия, визуализация и отчетность);*

В качестве регламентирующей документации представлены:

* *ГОСТы;*

В качестве потоков механизмов будут использоваться такие средства как:

* *ЭВМ*;
* *Инженеры-проектировщики*;

Диаграмма IDEF0 (А0) изображена на рис. 2.

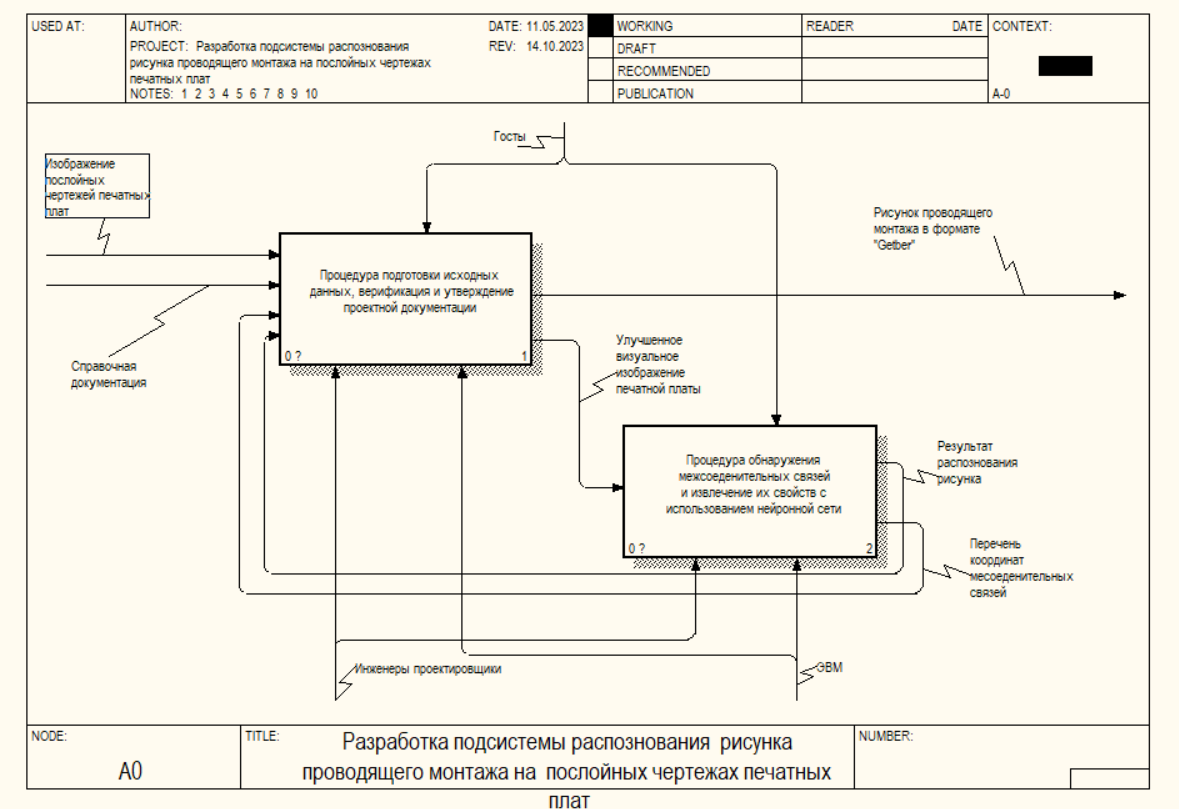


Рис.2 Контекстная диаграмма A0

**Описание IDEF0 A1 диаграммы**

1. Блок «Подготовка исходных данных». Данный блок отвечает за подготовку и обработку данных, необходимых для дальнейшего анализа и распознавания рисунка. Данный блок включает в себя преобразование данных в необходимы форматы и улучшение качества изображения при необходимости.

В качестве входных потоков информации представлены:

* *Изображение послойных чертежей печатных плат;*
* *Справочная документация;*

В качестве выходных потоков информации представлены:

* *Отредактированное изображение печатной платы;*

В качестве регламентирующей документации представлены:

* *ГОСТы;*

В качестве потоков механизмов будут использоваться такие средства как:

* *Инженеры-проектировщики*;
* *ЭВМ;*

1. Блок «Верификация и корректировка данных». Цель блока заключается в обеспечении высокой качественной информации, которая может быть использована в последующих принятиях решений. Это важный шаг помогает минимизировать ошибки и искажения, повышая достоверность и ценность данных. Данный блок включает в себя проверку правильности данных и их систематизированность, а также исправление выявленных недостатков.

В качестве входных потоков информации представлены:

* *Изображение послойных чертежей печатных плат;*
* *Перечень координат межсоединительных связей;*

В качестве выходных потоков информации представлены:

* *Отредактированное изображение печатной платы;*
* *скорректированные данные (результат выявления и исправления ошибок)*

В качестве регламентирующей документации представлены:

* *ГОСТы;*

В качестве потоков механизмов будут использоваться такие средства как:

* *Инженеры-проектировщики*;
* *ЭВМ;*

1. Блок «Подготовка технологического файла». Цель блока состоит в предоставлении структурированной, понятной и наглядной информации на рисунке. Данных блок включает в себя создание и подготовка файла с технологической информацией в формате “Gerber”.

В качестве входных потоков информации представлены:

* *скорректированные данные;*
* *Результат распознавания рисунка;*

В качестве выходных потоков информации представлены:

* *Рисунок проводящего монтажа в формате “Gerber”;*

В качестве регламентирующей документации представлены:

* *ГОСТы*

В качестве потоков механизмов будут использоваться такие средства как:

* *Инженеры-проектировщики*;

Диаграмма IDEF0 (А1) изображена на рис. 3.

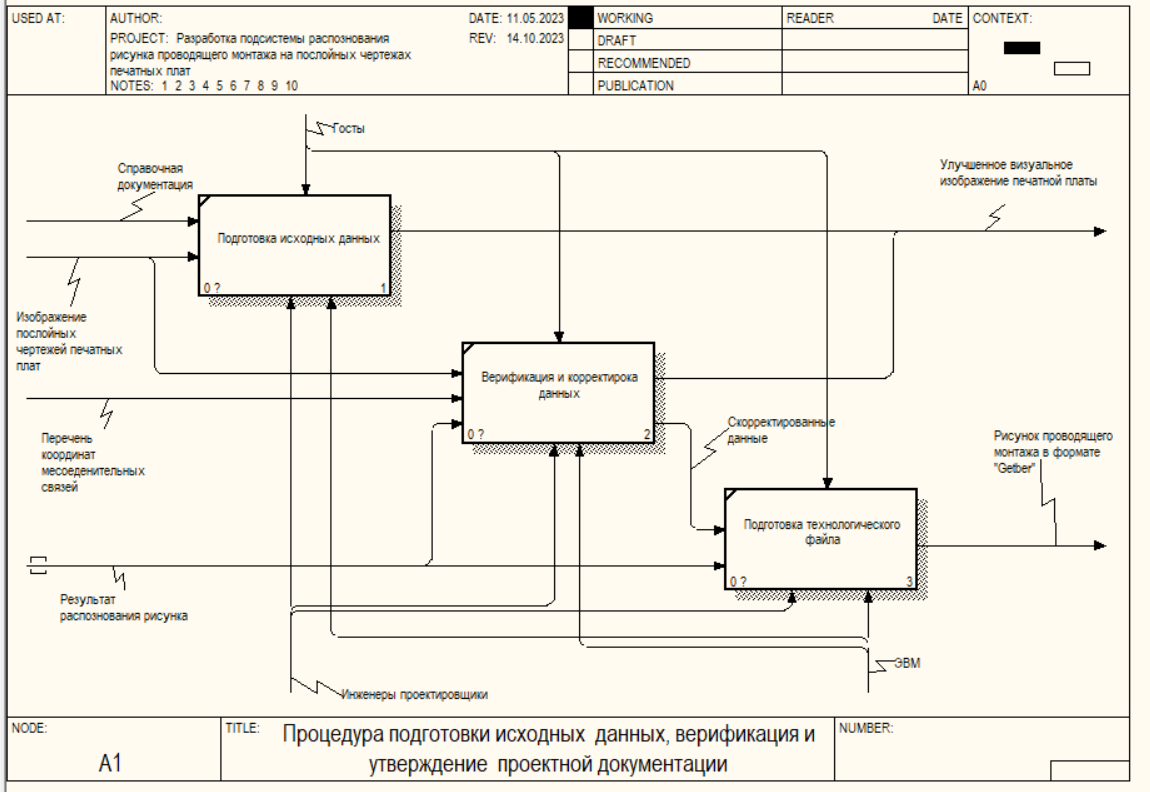


Рис.3 Контекстная диаграмма A1

**Описание IDEF0 A2 диаграммы**

1. Блок «Предварительная обработка изображения». Цель блока "Предварительная обработка изображения" состоит в подготовке изображений для дальнейшего анализа или обработки компьютерным зрением. Это позволяет улучшить качество данных, устранить шум, нормализовать характеристики изображений и выделить интересующие объекты для более точного и эффективного анализа или распознавания.

В качестве входных потоков информации представлены:

* *Отредактированное изображение печатной платы;*

В качестве выходных потоков информации представлены:

* *Отфильтрованное нормализованное изображение печатной платы* (изображение печатной платы после наложение таких способов коррекции как: фильтрация и улучшение изображений, нормализация и стандартизация, удаление фонов и сегментация)*;*

В качестве регламентирующей документации представлены:

* *ГОСТы*

В качестве потоков механизмов будут использоваться такие средства как:

* *Инженеры-проектировщики*;
* *ЭВМ.*

2. Блок «Пост-обработка данных». Цель блока "Пост-обработка данных" заключается в обеспечении более глубокого и детального анализа данных после их получения или первичной обработки. Данный блок может включать в себя удаление лишних элементов, объединение некоторых объектов и другие подобные операции, направленные на повышение читаемости изображения нейронной сетью.

В качестве входных потоков информации представлены:

* *Отфильтрованное нормализованное изображение печатной платы;*

В качестве выходных потоков информации представлены:

* *Изображение в требуемом нейросетью формате (*преобразование и изменение размера и формата изображения)

В качестве регламентирующей документации представлены:

* *ГОСТы;*

В качестве потоков механизмов будут использоваться такие средства как:

* *ЭВМ;*

3.Блок «Распознавание рисунка проводящего монтажа». Цель блока заключается в реализации процесса анализа и интерпретации рисунка проводящего монтажа нейронной сетью и предоставление информации о поданном на вход изображении.

В качестве входных потоков информации представлены:

* *Изображение в требуемом нейросетью формате;*

В качестве выходных потоков информации представлены:

* *Результат распознавания рисунка (*Визуализированные идентифицированные трассы проводящего монтажа и визуализация результатов)*;*

В качестве регламентирующей документации представлены:

* *ГОСТы;*

В качестве потоков механизмов будут использоваться такие средства как:

* *ЭВМ;*

Диаграмма IDEF0 (А2) изображена на рис. 4.

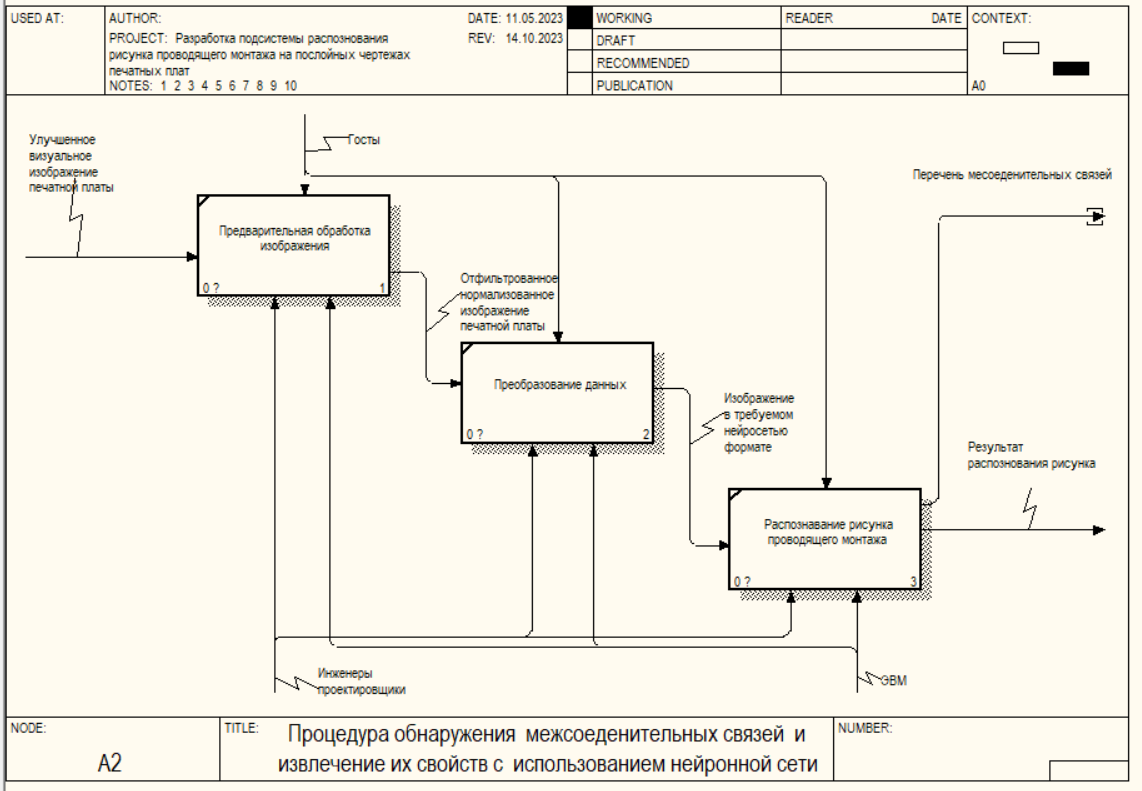


Рис.4 Контекстная диаграмма A2.

**1.2. Разработка поведенческих моделей (IDEF3)**

IDEF3 (Integrated Definition for Process Description Capture Method) - это методология, разработанная в США для описания бизнес-процессов. Она используется для моделирования и документирования процессов, связанных с информационными системами.

IDEF3 предназначена для описания процессов на уровне функциональной деятельности организации. Это значит, что методология позволяет описать, как выполняются определенные функции внутри организации, какие данные используются и какие результаты получаются.

IDEF3 использует графическую нотацию для описания процессов. Она состоит из блоков, которые представляют собой функции или действия, выполняемые в рамках процесса. Блоки соединяются линиями, которые показывают последовательность выполнения действий.

Одним из преимуществ IDEF3 является возможность использования этой методологии в сочетании с другими методами моделирования, такими как UML и BPMN. Это позволяет создавать более подробные и точные модели процессов.

Описание диаграммы (рис. 1). Для подготовки исходных данных процесса необходимо получить чертежи печатных плат и справочную документацию. В процессе подготовки определяется возможность работы с данными чертежами, а именно рассматривается соответствие всех составляющих чертежей согласно ГОСТу. Затем происходит подготовка изображения к работе, то есть обработка изображения, чтобы межсоединительные связи различались как можно лучше. После этого производится распознавание и построение общего рисунка монтажа электрических соединений, результат которого проходит верификацию. Далее идет этап на котором определяется подготовка описания и построение изображения. После этого проверяется правильность построение изображения и корректировка данных. На последнем этапе мы готовим рисунок проводящего монтажа в формате “Gerbrer”.

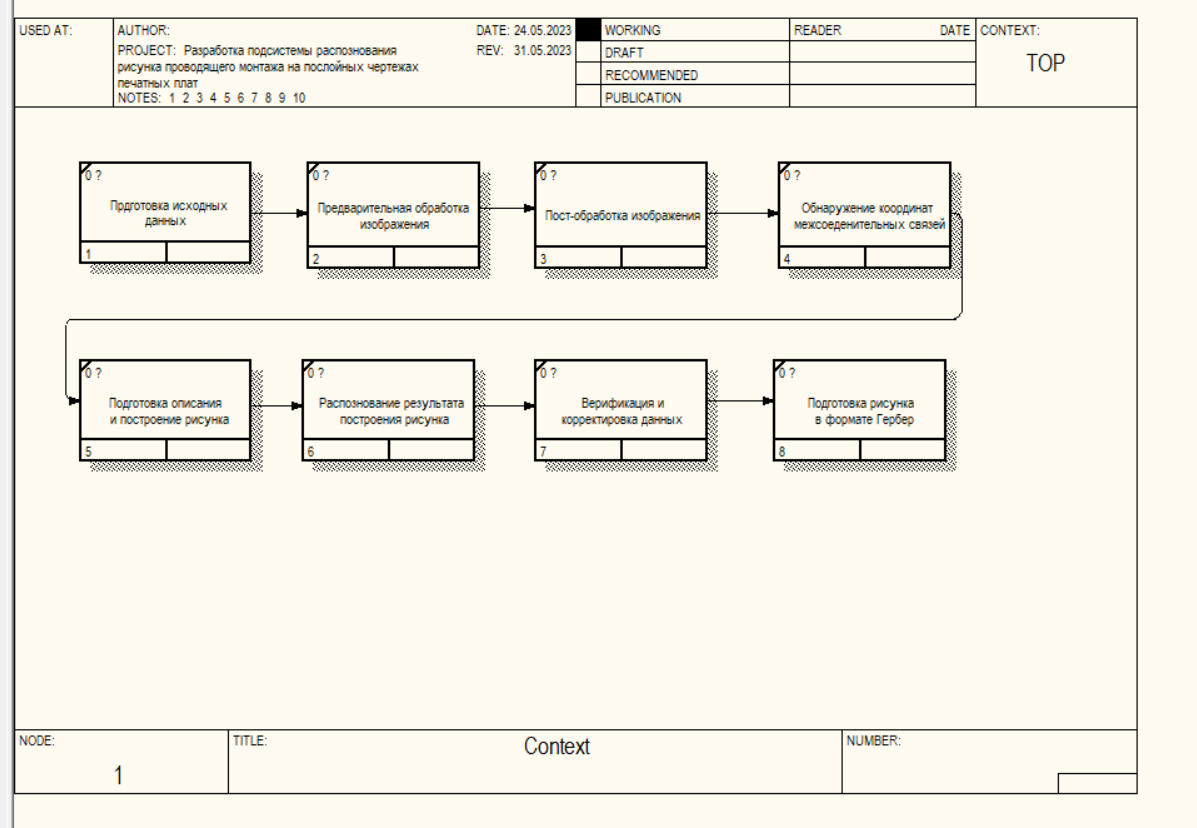


Рис. 5. Диаграмма IDEF3

**Глава 2. Разработка математического обеспечения.**

**2.1. Содержательная постановка задачи**

Для разработки подсистемы распознавания рисунка проводящего монтажа на послойных чертежах печатных плат необходимо собрать набор послойных чертежей печатных плат, содержащих рисунки проводящего монтажа.

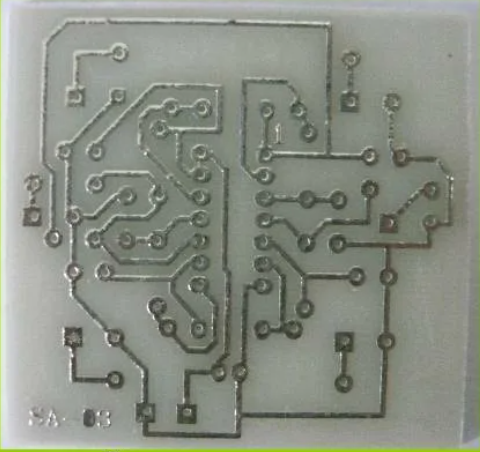


Рис.6 Пример печатной платы, содержащей проводящий монтаж

Данные должны быть подготовлены и представлены в удобном формате для дальнейшего анализа и обработки. Возможно потребуется предварительная обработка изображений для улучшения их качества, удаления шума, коррекции и улучшения контрастности или яркости. Это поможет повысить точность распознавания и улучшить результаты.

На этапе подготовки данных нам необходимо разделить изображения выборки на тренировочную и тестовую. Распределение выборки на тренировочную и тестовую части является важным этапом в процессе подготовки данных для обучения моделей нейронных сетей. Этот процесс позволяет оценить эффективность модели после её обучения и определить её способность к обобщению полученных знаний.

Неотъемлемой частью задачи разработки подсистемы распознавания рисунка проводящего монтажа на послойных чертежах печатных плат является - построение модели распознавания элементов на изображении.

Построение модели для задачи распознавания объектов на изображении включает в себя создание алгоритма или нейронной сети, способных локализовать и классифицировать объекты на входном изображении. Для этого используются различные архитектуры, такие как Faster R-CNN, YOLO, SSD и другие.

Модель обучается на размеченном наборе данных, где кроме самих объектов указываются их координаты (ограничивающие рамки) и классы. В процессе обучения модель настраивает свои веса, стремясь минимизировать функцию потерь, что позволяет ей точнее определять и классифицировать объекты на новых изображениях.

Критическими этапами являются подготовка данных, выбор подходящей архитектуры модели, оптимизация гиперпараметров, а также оценка качества модели на тестовых данных. После обучения, модель может быть интегрирована в решение, где будет применяться к реальным изображениям для задачи распознавания объектов.

Также для полной автоматизации процесса распознавания рисунка проводящего монтажа на послойных чертежах печатных плат, потребуется создание и управление базой данных компонентов, содержащей информацию о размеченных данных. Также база данных послужит хранилищем для размеченных моделью выходных изображений печатных плат.

Результатом работы подсистемы должен быть размеченное изображение проводящего монтажа и анализ связей схемы и интерпретация данных распознавания рисунка проводящего монтажа. Конечный распознанный рисунок проводящего монтажа должен включать в себя проверку правильности распознавания компонентов, сопоставление справочным данным, анализ структуры и логики монтажа, а также обнаружение потенциальных ошибок или несоответствий.

Цель разработки подсистемы распознавания рисунка проводящего монтажа на послойных чертежах печатных плат заключается в создании эффективного и автоматизированного инструмента для анализа и работы с данными проводящего монтажа, что способствует повышению производительности, точности и качества процесса проектирования и производства печатных плат.

**2.2. Математическая постановка задачи.**

Математическая постановка задачи разработки подсистемы распознавания рисунка проводящего монтажа на послойных чертежах печатных плат сводится к задаче построения модели распознавания объектов на изображении.

Модель распознавания -это компьютерная алгоритмическая система, предназначенная для автоматического обнаружения и локализации объектов определенного класса или категории на изображении или в видеопотоке. Она позволяет выделить интересующие нас элементы сцены и, часто, также указать их ограничивающие рамки (bounding boxes) для наглядности.

Пусть С представляет собой чертежи печатной платы, где каждый чертеж представлен в виде двумерного массива пикселей. Тогда каждое изображение имеет размерность Сi - [YxZ]. При этом каждый пиксель P имеет собственную цветовую гамму, которая рассчитывается в диапазоне от 0 до 255 Pi -(0...255)

При построении модели M распознавания элементов на изображении также необходимо рассчитать количество слоев S и нейронов на каждом слое N.

Каждый слой будет иметь вид - Si[N]. А суммарное количество нейронов N будет вычисляться по формуле: N =   
 До начала обучения нейронной сети, веса θ у всех нейронов N устанавливаются случайным образом. Это начальное состояние называется "инициализацией весов". Правильный выбор начальных весов может существенно повлиять на процесс обучения и качество модели.

Инициализация весов θ важна, потому что она задает отправную точку для оптимизационного процесса. Веса определяют, как входные данные будут взвешиваться и влиять на активацию нейронов.

Случайная инициализация весов θ обычно осуществляется с использованием небольших случайных чисел, часто из нормального распределения с нулевым средним (mean) и небольшой стандартной дисперсией (standard deviation). Это позволяет сети начать процесс обучения с разнообразными весами, что помогает избежать симметрии и ускоряет сходимость.

Результатом данной задачи является изображение Y2 с выделенными необходимыми элементами.

Для математической (формальной) постановки задачи, введём следующие обозначения, описанные в таблице 1 ниже.

Таблица 1. Перечень обозначений элементов в математической модели задачи.

|  |  |  |
| --- | --- | --- |
| **Обозначения элементов математической модели** | **Описание соответствующих элементов математической модели** | **Примечание** |
| f(x; θ) | Нейронная сеть | где x(i) - вектор входных признаков для i-го объекта, а θ - вектор параметров (весов) сети |
| θ | Вес нейрона | В большинстве случаев является случайной величиной |
| L(θ) | Функция потерь |  |
| Сi - [YxZ] | Послойный чертеж печатной платы из множества С | Каждый Сi является матрицей размером Y x Z, где Y - количество пикселей по вертикали, Z - количество пикселей по горизонтали. |
| Y1i[P] | Входное изображение печатной платы | P - количество пикселей на изображении |
| Pi -(0...255) | Значение цвета пикселя изображении Xi | Цветовая гамма пикселя програмно ограничивается в диапазоне от 0 до 255. |
| M[S] | Модель распознавания | Модель распознавания состоит из слоев и нейронов на каждом из слоев, где S - количество слоев |
| Si[N] | Слой модели распознавания | Si состоит из нейронов N |
| Y2i[P] | Изображение после прохождение через модель распознавания | Выходное изображение имеет ту же размерность, что и входное |

**2.3. Разработка модели для задачи** **распознавания рисунка проводящего монтажа на послойный чертежах печатных плат**

Для решения задачи построения модели распознавания рисунка проводящего монтажа на послойный чертежах печатных плат необходимо применяются сверточные нейронные сети (CNN), специально разработанные для анализа изображений. Они обучаются выделять признаки элементов проводящего монтажа, такие как контуры, формы и размеры.

Однако, чтобы модель работала оптимально, необходимо правильно подобрать гиперпараметры нейросети. Здесь применяется автоматическая оптимизация, позволяющая найти оптимальные значения параметров с учетом конкретной задачи.

Результаты данной разработки представляют собой эффективную систему, способную автоматически распознавать рисунок проводящего монтажа на послойных чертежах печатных плат. Это значительно упрощает производственный процесс и повышает качество готовой продукции.

На данном этапе я постарался выбрать наиболее эффективную и одновременно менее ресурсоемкую модель. Для подбора гиперпараметров были выбраны такие критерии как: количество слоев, количество нейронов на каждом слое и функция активации.

После ряда проведенных операций, эксперементным путем было выяснено что наиболее результативная функция активации - relu; количество слоев - 4 сверточных слоя, 2 – слоя подвыборки, 3 –слоя регуляризации, cлой преобразования данных из 2D представления в плоское и выходной слой. На некоторых слоях модели применялись константные значения нейронов для упрощения и ускорения процесса подбора гиперпараметров.

**2.4. Описание алгоритма обучения нейронной сети.**

Для выполнения задачи обучения модели был выбран метод градиентного спуска.

Обучение модели нейронной сети с помощью регрессии можно представить в виде следующей математической постановки задачи:

* + Заданы обучающие данные: набор пар (x(i), y(i)), где x(i) - вектор входных признаков для i-го объекта, y(i) - целевой вещественный скаляр для i-го объекта.
  + Нейронная сеть представлена в виде функции f(x; θ), где θ - вектор параметров (весов) сети. Функция f представляет собой композицию нескольких функций (слоев) и имеет несколько скрытых слоев.
  + Функция потерь (objective function) определяется как сумма квадратов ошибок между предсказанными значениями сети f(x(i); θ) и целевыми значениями y(i):
  + L(θ) = 1/2N ∑i=1N (f(x(i);θ) - y(i))^2
  + Задача обучения модели состоит в том, чтобы найти такие значения параметров θ, которые минимизируют функцию потерь L(θ).
  + Для решения этой задачи используется метод градиентного спуска (или его модификации), который заключается в итеративном обновлении параметров θ в направлении антиградиента функции потерь L(θ):

θ(t+1) = θ(t) - α \* ∇θL(θ(t))

где α - это шаг градиентного спуска.

Процесс обучения продолжается до тех пор, пока функция потерь не достигнет некоторого заранее заданного значения или не перестанет существенно уменьшаться на протяжении нескольких итераций.

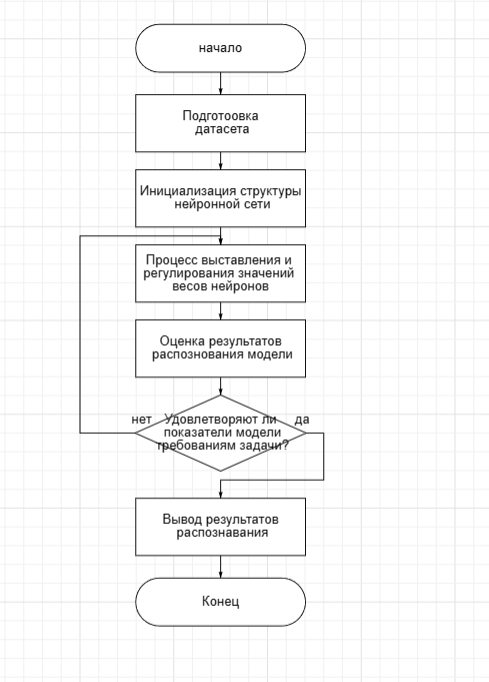


Рис. 7 – Блок-схема метода обучения нейронной сети

**2.4. Решение задачи на контрольном примере.**

Рассмотрим контрольный пример разработки подсистемы распознавания рисунка проводящего монтажа на послойных чертежах печатных плат с использованием нейронной сети.



Рис. 8. Контрольный пример #1

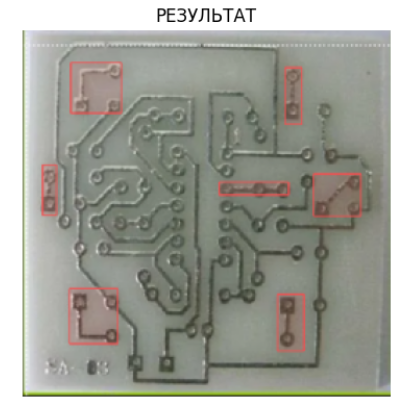


Рис. 9. Решение контрольного примера #1



Рис. 10. Контрольный пример #2

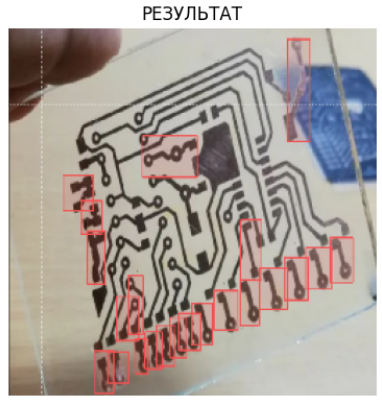


Рис. 11. Решение контрольного примера #2



Рис. 12. Контрольный пример #3

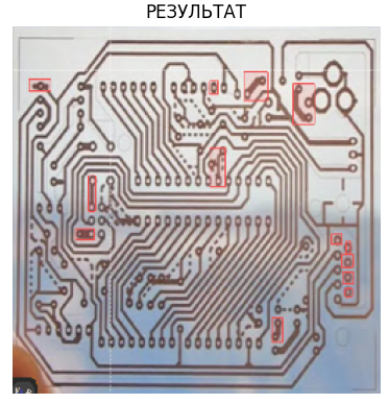


Рис. 13. Решение контрольного примера #3

Точность результатов распознавания зависит от многих факторов, включая качество исходных данных, сложность распознаваемого объекта. Результаты распознавания кажутся недостаточно точными, но почти безошибочными. Для обеспечения более высокой точности необходимо иметь больший объем выборки и лучшее качество исходных данных(изображений печатных плат), а также может помочь разработка более сложной модели распознавания.

**Глава 3. Разработка информационного обеспечения.**

**3.1 Разработка модели потока данных (DFD) проектной процедуры распознавания рисунка проводящего монтажа на послойный чертежах печатных плат**

**Модель DFD**

Диаграммы потоков данных (Data Flow Diagrams — DFD) представляют собой иерархию функциональных процессов, связанных потоками данных. Цель такого представления — продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

Модель окружения (Рис. 1) описывает систему как определенный объект, отвечающий на события, которые порождаются внешними сущностями. Функциональный блок с основным процессом “ Распознавание проводящего монтажа на послойных чертежах печатных плат” который обозначает моделируемую систему, соединен с информационными связями внешних сущностей.

В качестве входных источников данных:

* Инженер-конструктор – это инженер, ответственный за подготовку изображений и их анализ.

В качестве выходных источников данных:

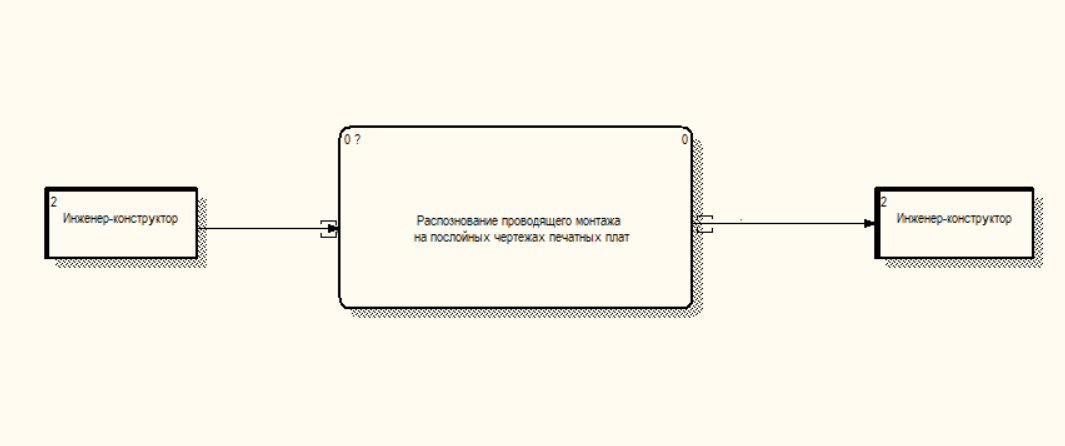
* Инженер-конструктор – это инженер, ответственный за верификацию и утверждение проектной документации;
* 

Рис. 14 Модель окружения

Логическая модель (Рис. 2) представляет систему в виде набора действий и определяет порядок выполнения этих действий. Она описывает систему в терминах функциональных блоков, которые взаимодействуют между собой через потоки данных. Что позволяет представить систему в виде графической диаграммы, где блоки представляют отдельные функции, а потоки данных показывают передачу информации и результатов между блоками.

Основные процессы (функции) в моделируемой системе:

“Подготовка изображения” – выбор чертежа и обработка изображения, входными потоками данных является:

* Чертёж печатной платы;
* Перечень УГО проводящего монтажа;

Выходным потоком является обработанное изображение принципиальной электрической схемы.

“Предварительная обработка изображения” – этап обработки изображения состоит в подготовке рисунков проводящего монтажа для дальнейшего анализа или обработки компьютерным зрением. Это позволяет улучшить качество данных, устранить шум, нормализовать характеристики изображений и выделить интересующие объекты для более точного и эффективного анализа или распознавания.

“Пост-обработка данных из”. "Пост-обработка данных изображения" заключается в обеспечении более глубокого и детального анализа данных после их получения или первичной обработки. Данный блок может включать в себя удаление лишних элементов, объединение некоторых объектов и другие подобные операции, направленные на повышение читаемости изображения нейронной сетью.

“Распознавание рисунка проводящего монтажа” - это задача компьютерного зрения, которая заключается в разделении изображения на отдельные сегменты или регионы, содержащие схожие характеристики или объекты.

Выходным потоком данных является сегментированное изображение печатной платы.

“Верификация технической информации” - обеспечивает контроль качества работы системы распознавания, что позволяет повысить точность и надежность распознавания технической информации на послойных чертежах печатных плат.

“Подготовка технического файла” состоит в формировании и подготовке данных для работы подсистемы распознавания, что обеспечивает точность и эффективность распознавания рисунка проводящего монтажа на чертежах печатных плат.

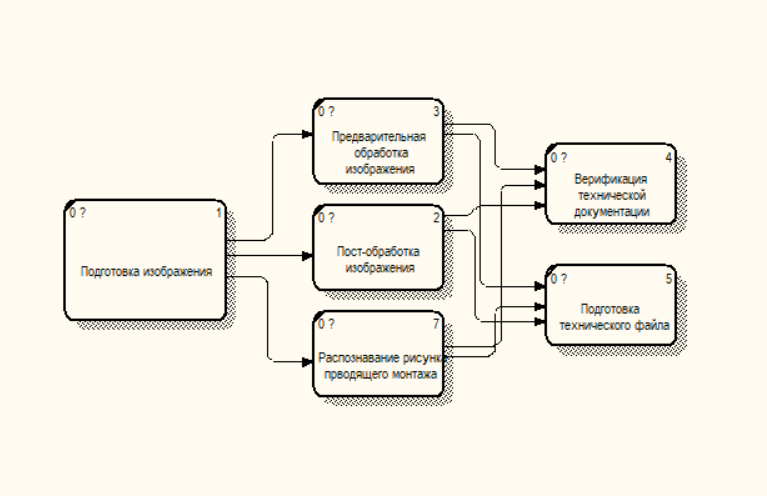


Рис. 15 Логическая модель

Модель поведения (Рис. 3) описывает последовательность обработки событий в системе. Она включает в себя внешние сущности, которые определены в модели окружения и выступают в роли источников и/или приемников информации, а также функциональные блоки, которые определены в логической модели и отвечают за обработку этой информации.

Модель поведения представляет собой описание того, как события и данные передаются между внешними сущностями и функциональными блоками, позволяя показать порядок их взаимодействия и обработки.

Взаимосвязь между блоками отражает передачу и прием информации. Чтобы учесть процессы сохранения данных, применяются хранилища данных. В результате, модель поведения представляет собой комплексную схему, которая дополняет и расширяет логическую модель и модель окружения.

Хранилище данных — это абстрактное устройство или способ хранения информации, которая перемещается между различными процессами.

Хранилища данных, которые служат входными данными для блока процесса “ Распознавание проводящего монтажа”:

* Чертёж печатной платы, разработанный инженером-конструктором;

Хранилища данных, которые служат выходными данными для блока процесса “Пред обработки изображения”, “Пост-обработки изображения”, “Распознавание проводящего монтажа изображения”:

* Перечень координат проводящего монтажа – это результат распознавания межсоеденительных связей на рисунке и их представления в формате (x,y)
* Рисунок связей

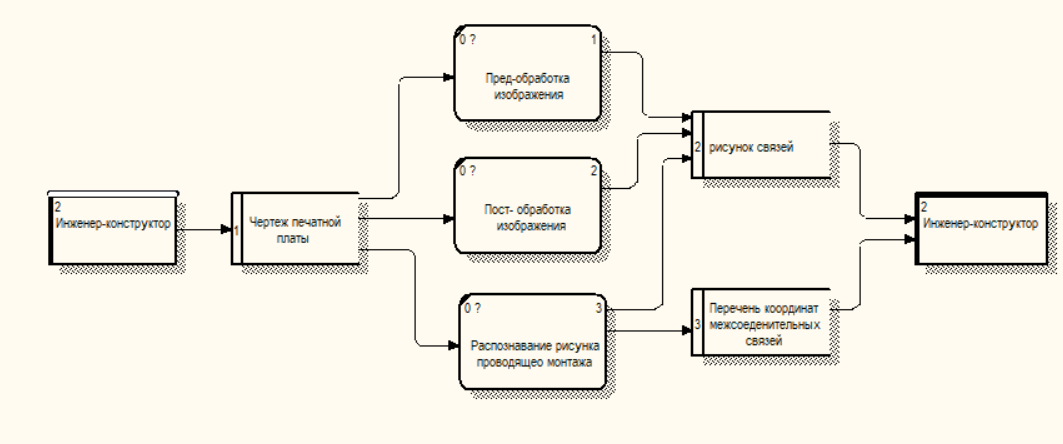


Рис. 16 Модель поведения

**3.2 Разработка концептуальной модели базы данных**

Рассмотрим сущности проектируемой базы данных ниже в таблице 2.

Таблица 2. Сущности базы данных

|  |  |  |  |
| --- | --- | --- | --- |
| № | Обозначение сущности | Наименование сущности | Количество экземпляров |
| 1 | S1 | Входные тестовые изображения | 9 |
| 2 | S2 | Входные тестовые размеченные изображения | 9 |
| 3 | S3 | Входные тренировочные изображения | 30 |
| 4 | S4 | Входные тренировочные размеченные изображения | 30 |
| 5 | S5 | Изображения для разметки | 7 |
| 6 | S6 | Изображения содержащие результаты распознавания | 7 |
| 7 | S7 | Модель нейронной сети | 1 |

“Входные тестовые изображения” - это сущность, хранящая в себе информацию о изображениях, которые необходимы для проверки результатов обучения.

“Входные тестовые размеченные изображения” - это сущность, хранящая в себе информацию о промаркированных изображениях, которые необходимы для проверки результатов обучения.

“Входные тренировочные изображения” - это сущность, хранящая в себе информацию о изображениях, которые необходимы для обучения модели.

“Входные размеченные тренировочные изображения” - это сущность, хранящая в себе информацию о промаркированных изображениях, которые необходимы для обучения модели.

“Изображения для разметки” - это сущность, хранящая в себе информацию о изображениях, которые должны быть обработаны в качестве контрольного примера.

“Изображения содержащие результаты распознавания” - это сущность, хранящая в себе информацию о изображениях, которые были обработаны в качестве контрольного примера и содержат информацию о проводящем монтаже.

“Модель нейронной сети” - это сущность, хранящая в себе информацию о конфигурации модели распознавания рисунка проводящего монтажа на послойный чертежах печатных плат.

Таблица 3. Атрибуты и домены, относящиеся к сущностям

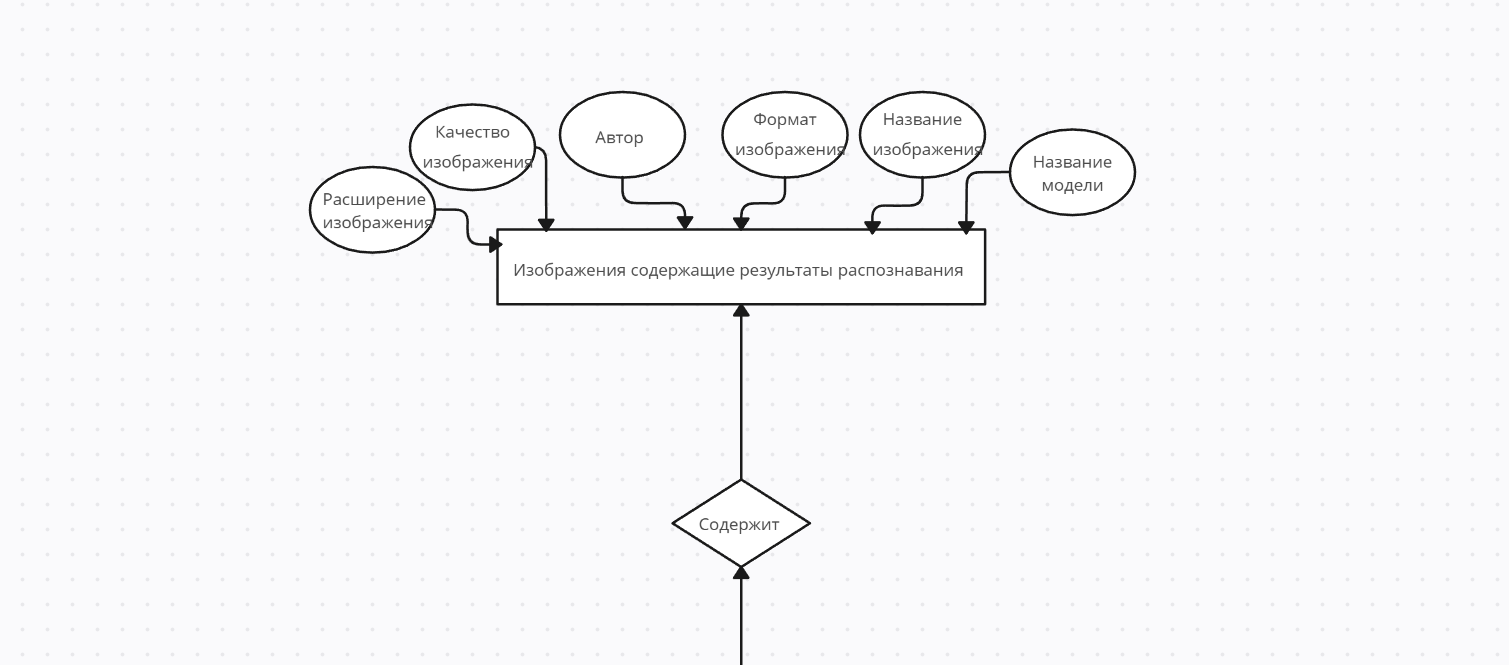
|  |  |  |
| --- | --- | --- |
| Название сущности | Название атрибута | Наименование домена |
| Входные тестовые изображения | Название изображения | Название |
| Название каталога | Название |
| Автор | Личное имя |
| Формат изображения | Название формата |
| Качество изображения | Качество |
| Расширение изображения | Название расширения |
| Входные тестовые размеченные изображения | Название изображения | Название |
| Название каталога | Название |
| Автор | Личное имя |
| Формат изображения | Название формата |
| Качество изображения | Качество |
| Расширение изображения | Название расширения |
| Входные тренировочные изображения | Название изображения | Название |
| Название каталога | Название |
| Автор | Личное имя |
| Формат изображения | Название формата |
| Качество изображения | Качество |
| Расширение изображения | Название расширения |
| Входные размеченные тренировочные изображения | Название изображения | Название |
| Название каталога | Название |
| Автор | Личное имя |
| Формат изображения | Название формата |
| Качество изображения | Качество |
| Расширение изображения | Название расширения |
| Изображения для разметки | Название изображения | Название |
| Название каталога | Название |
| Автор | Личное имя |
| Формат изображения | Название формата |
| Качество изображения | Качество |
| Расширение изображения | Название расширения |
| Изображения содержащие результаты распознавания | Название изображения | Название |
| Название каталога | Название |
| Автор | Личное имя |
| Формат изображения | Название формата |
| Качество изображения | Качество |
| Расширение изображения | Название расширения |
| Модель нейронной сети | Название каталога | Название |
| Автор | Личное имя |
| Название модели | Название |
| Количество слоев модели | Число |
| Количество нейронов модели | Число |

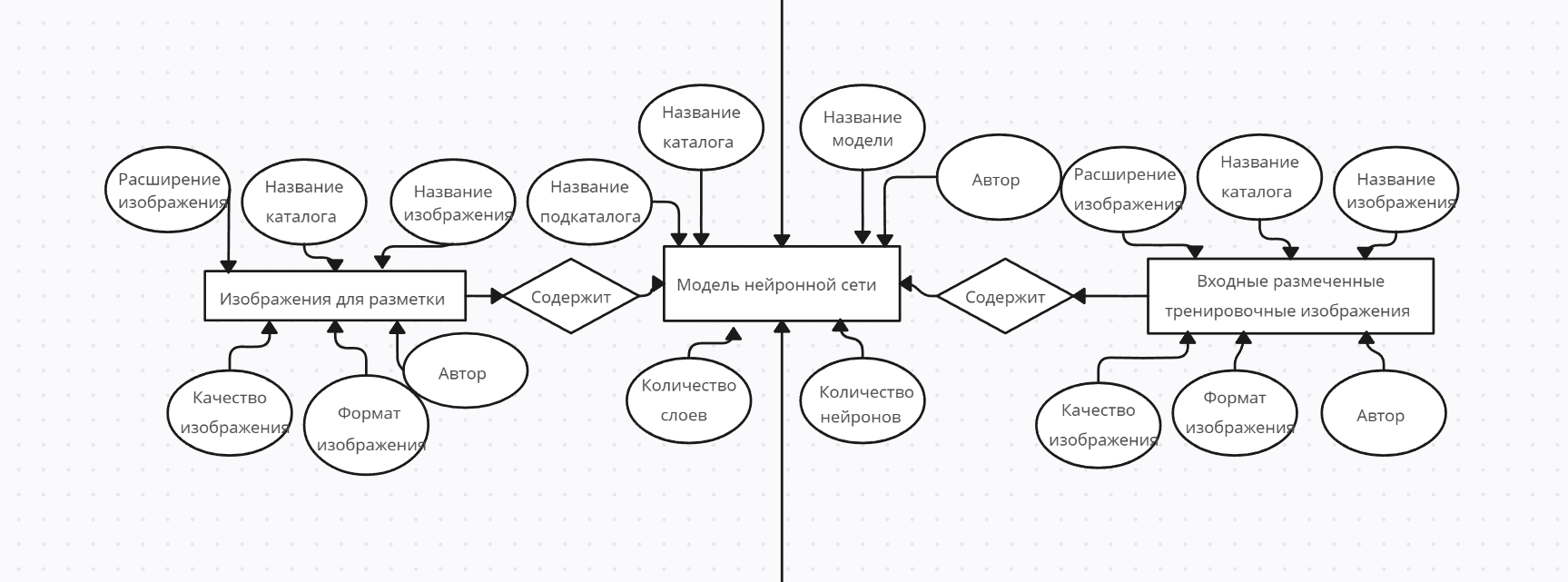
Для доменов определяются типы значений в таблице 4.

Таблица 4. Типы значений доменов

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название домена | Наименование и тип данных | Пример значений |
| 1 | Название изображения | Строка символов, цифр и пробелов | Marked\_image\_test\_1.jpeg |
| 2 | Название каталога | Строка символов, цифр и пробелов | test/Marked |
| 3 | Автор | Фамилия и инициалы в формате ГОСТ Р 7.0.97–2016 | Бикмуллин А.Р |
| 4 | Название модели | Строка символов, цифр и пробелов | xmodel |
| 5 | Количество слоев модели | Целое число без пробелов | 10 |
| 6 | Количество нейронов модели | Целое число без пробелов | 800 |
| 7 | Формат изображения | Строка символов | JPEG |
| 8 | Качество изображения | Строка символов, цифр и пробелов | 900p |
| 9 | Расширение изображения | Строка символов, цифр и пробелов | 250x250 |

На рисунке 17 ниже представлена концептуальная модель локальной области.





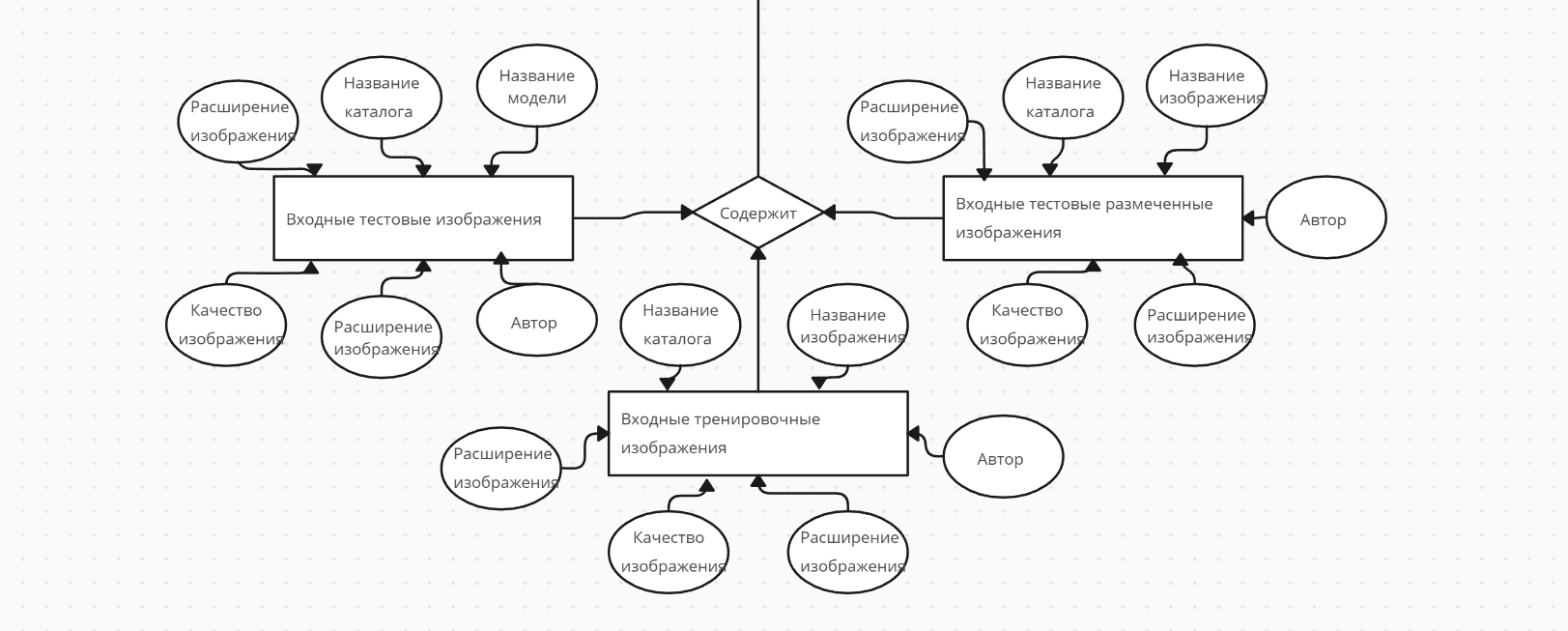


Рис. 17 Концептуальная модель локальной области

**3.3 Разработка логической модели базы данных**

Используя концептуальную модель, построим логическую модель базы данных. В таблице 5 представлена система функциональных зависимостей для каждого отношения из проектируемой базы данных.

Таблица 5. Система функциональных зависимостей.

|  |  |  |
| --- | --- | --- |
| Название сущности | Название атрибута | F1 |
| Входные тестовые изображения | Название изображения | \* |
| Название каталога | \* |
| Дата создания | ← |
| Дата изменения | ← |
| Автор | ← |
| Формат изображения | ← |
| Качество изображения | ← |
| Расширение изображения | ← |
| Входные тестовые размеченные изображения | Название изображения | \* |
| Название каталога | \* |
| Дата создания | ← |
| Дата изменения | ← |
| Автор | ← |
| Формат изображения | ← |
| Качество изображения | ← |
| Расширение изображения | ← |
| Входные тренировочные изображения | Название изображения | \* |
| Название каталога | \* |
| Дата создания | ← |
| Дата изменения | ← |
| Автор | ← |
| Формат изображения | ← |
| Качество изображения | ← |
| Расширение изображения | ← |
| Входные размеченные тренировочные изображения | Название изображения | \* |
| Название каталога | \* |
| Дата создания | ← |
| Дата изменения | ← |
| Автор | ← |
| Формат изображения | ← |
| Качество изображения | ← |
| Расширение изображения | ← |
| Изображения для разметки | Название изображения | \* |
| Название каталога | \* |
| Дата создания | ← |
| Дата изменения | ← |
| Автор | ← |
| Формат изображения | ← |
| Качество изображения | ← |
| Расширение изображения | ← |
| Изображения содержащие результаты распознавания | Название изображения | \* |
| Название каталога | \* |
| Дата создания | ← |
| Дата изменения | ← |
| Автор | ← |
| Формат изображения | ← |
| Качество изображения | ← |
| Расширение изображения | ← |
| Модель нейронной сети | Название каталога | \* |
| Дата создания | ← |
| Дата изменения | ← |
| Автор | ← |
| Название модели | \* |
| Количество слоев модели | ← |
| Количество нейронов модели | ← |

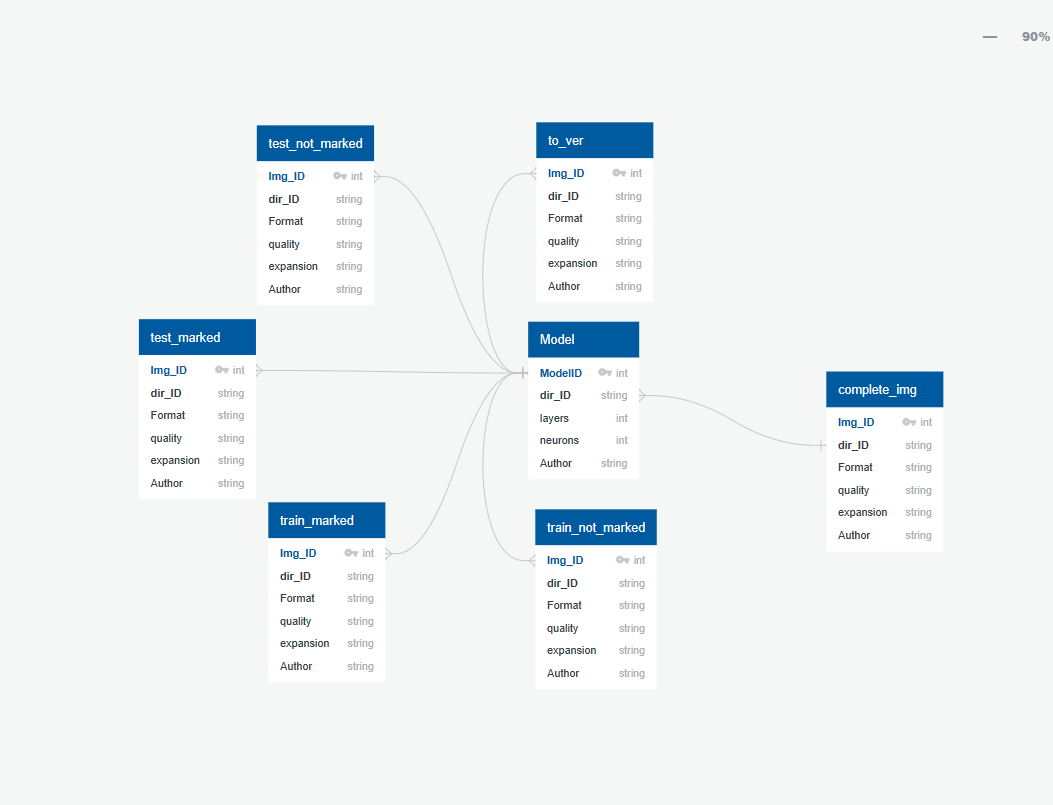


Рис. 18. Логическая модель базы данных

**3.4. Физическое проектирование базы данных**

Для физического проектирования базы данных и формирования запросов к ней была выбрана достаточно мощная и свободная СУБД PostgreSQL. Диаграмма базы данных, полученная в pgAdmin 4 представлена на рисунке 19.

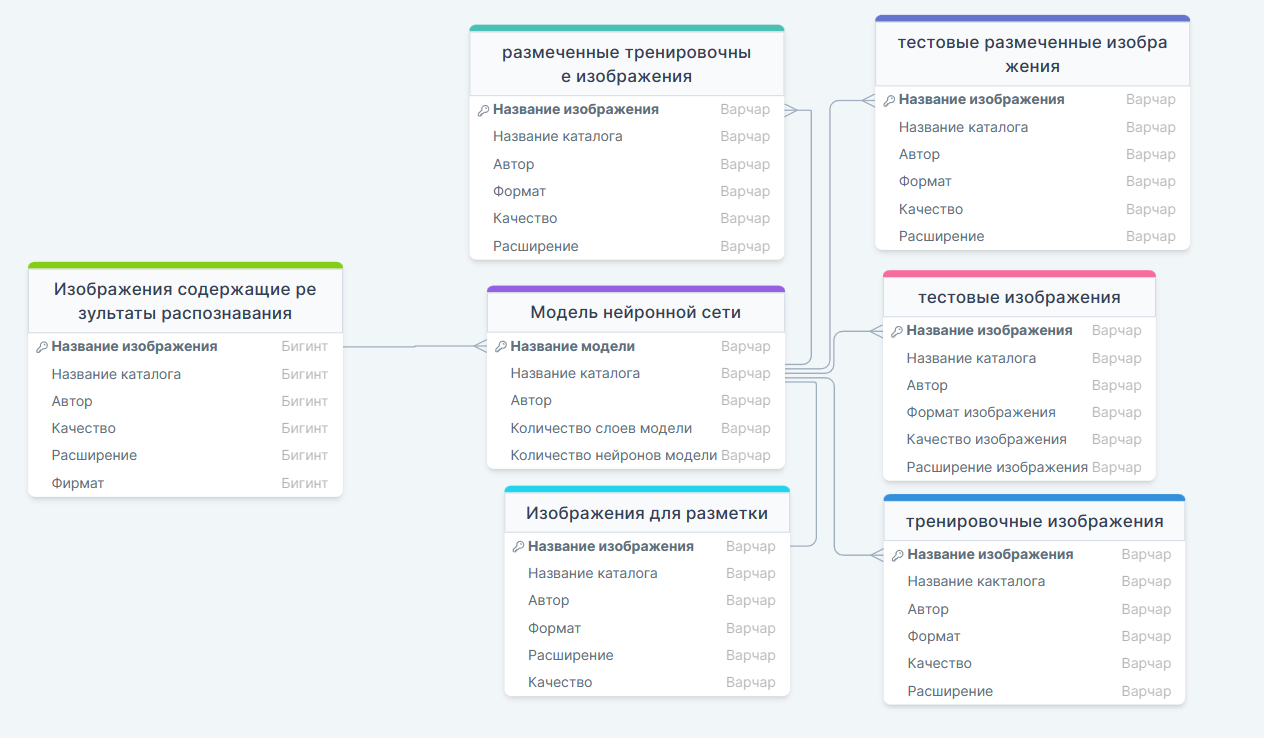


Рис. 19. Диаграмма баз данных

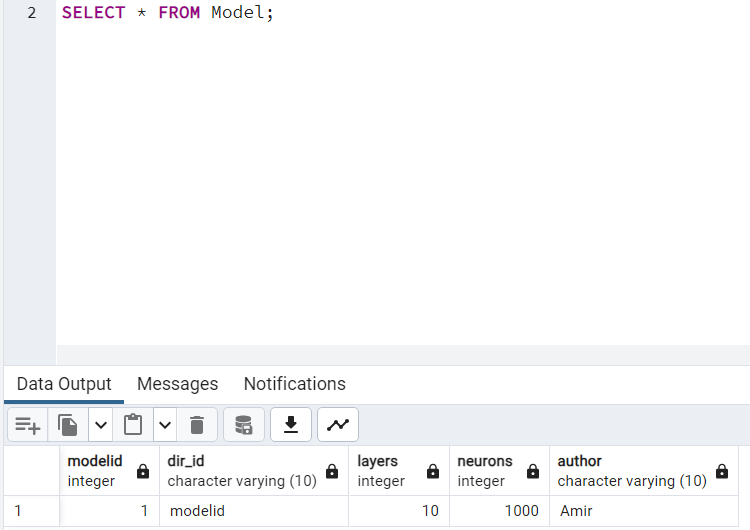


Рис. 20. Сущность «model»

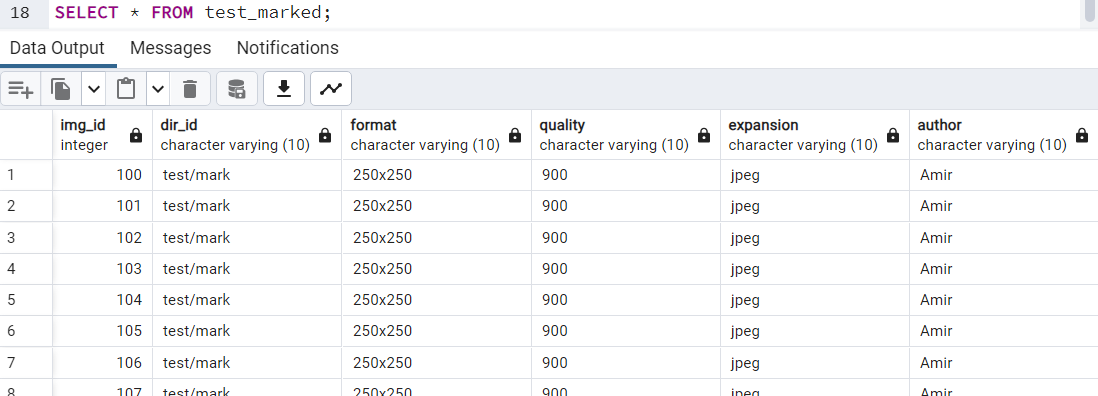


Рис. 21. Сущность «Входные тестовые размеченные изображения»

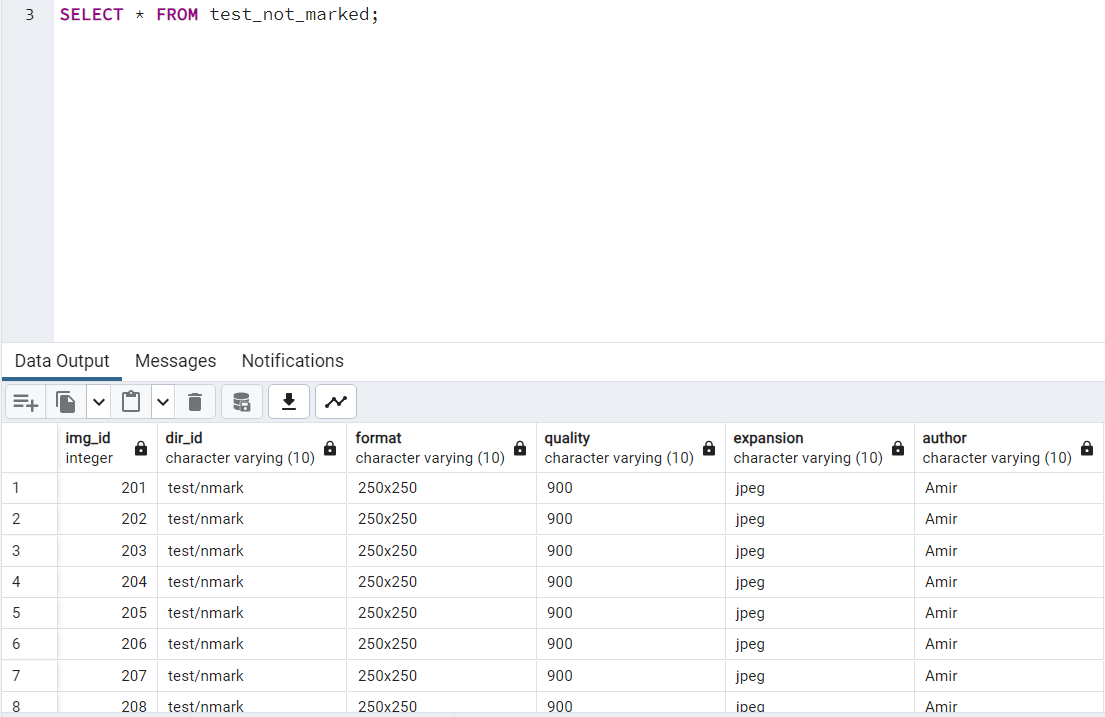


Рис. 22. Сущность «Входные тестовые изображения»

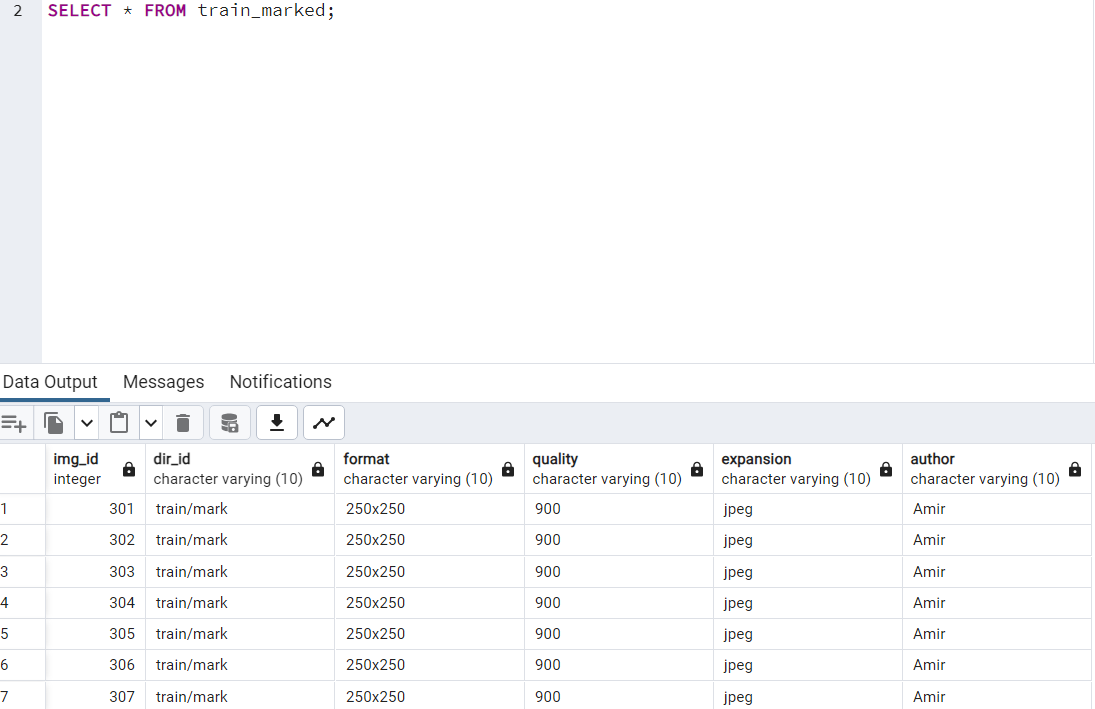


Рис. 23. Сущность «Входные тренировочные изображения»

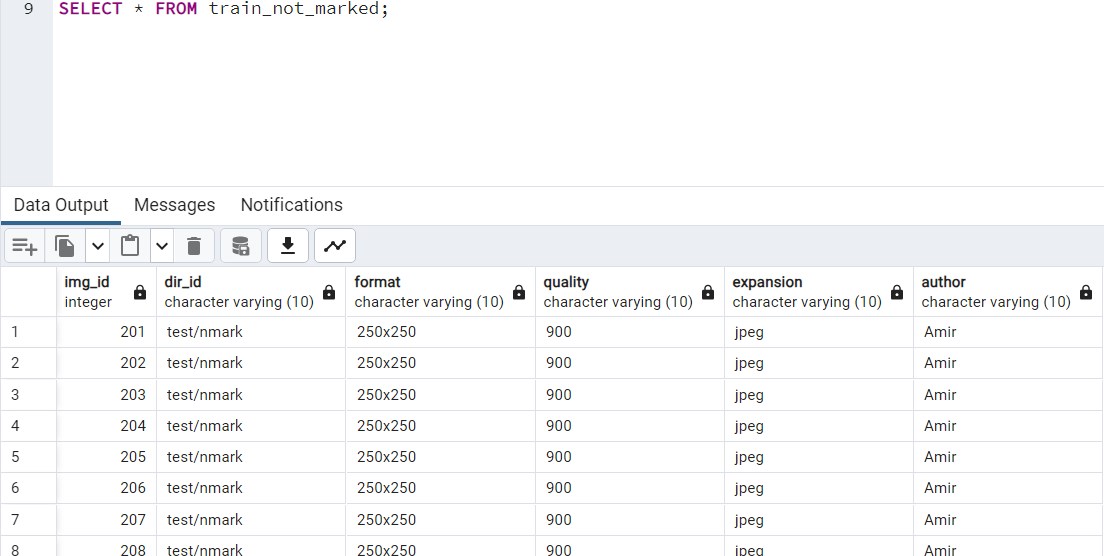


Рис. 24. Сущность «Входные тренировочные изображения»

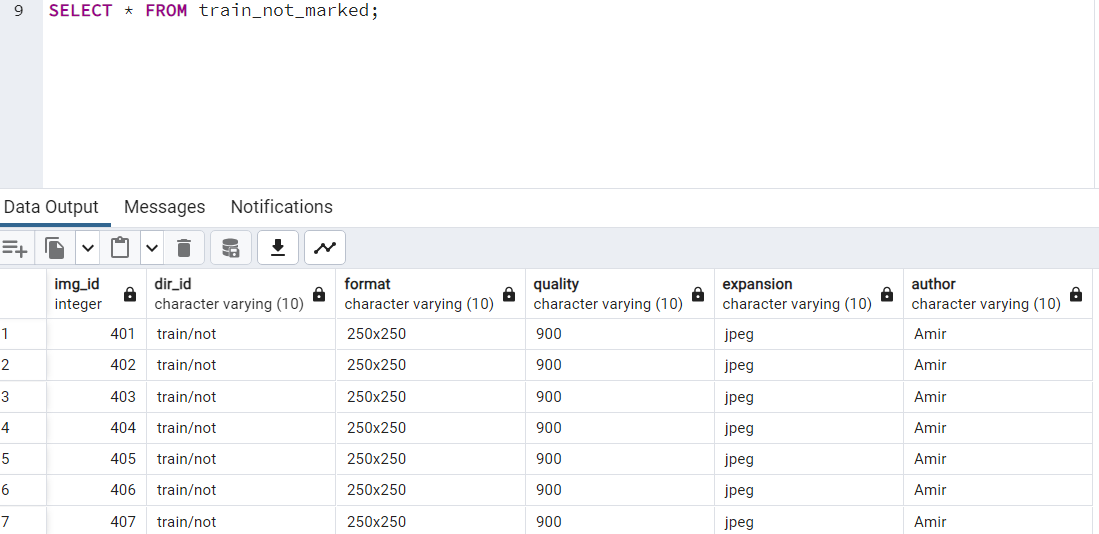


Рис. 25. Сущность «Входные размеченные тренировочные изображения»

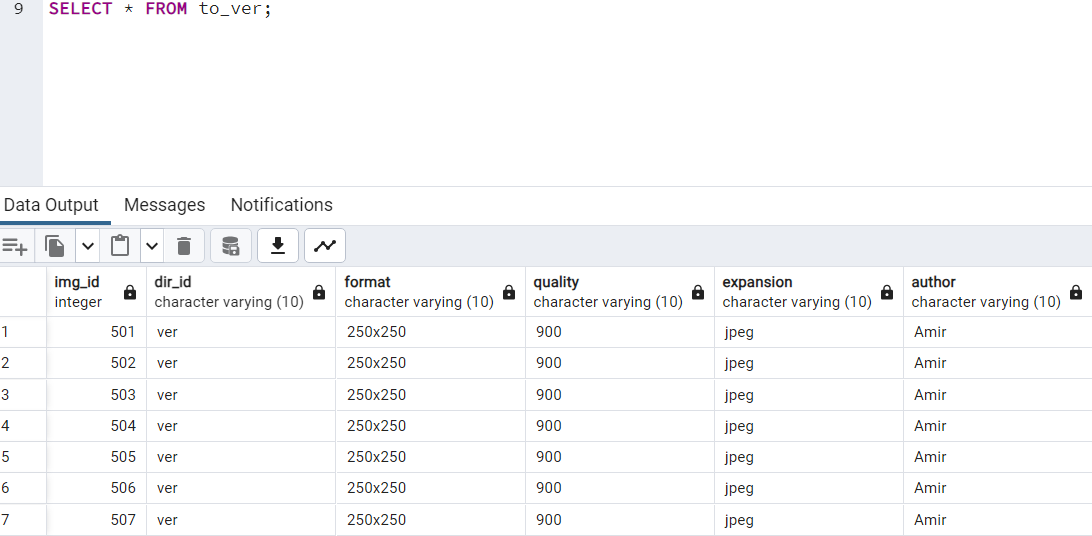


Рис. 26. Сущность «Изображения для разметки»

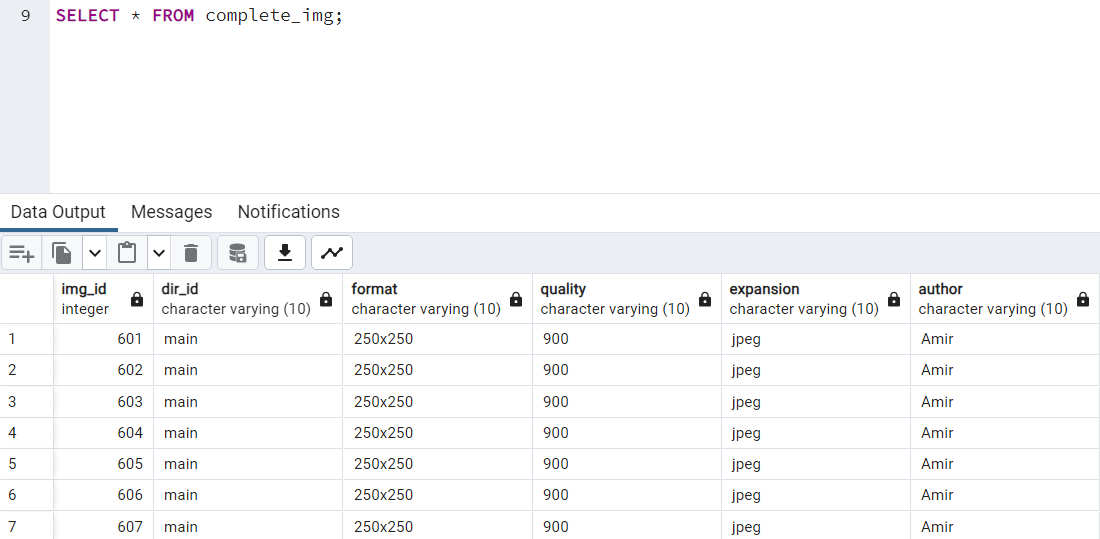


Рис. 27. Сущность «Изображения содержащие результаты распознавания»

Реализуем некоторые SQL запросы к базе данных.



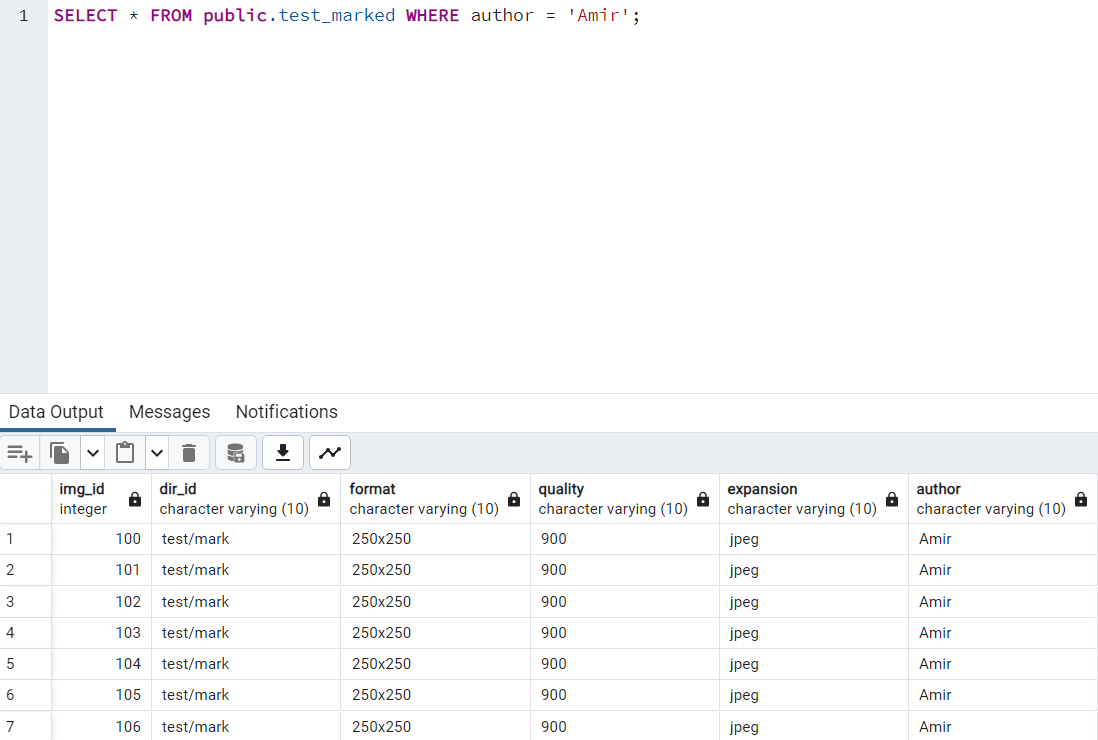
Рис 28. Пример SQL запроса

Рис 29. Пример SQL запроса

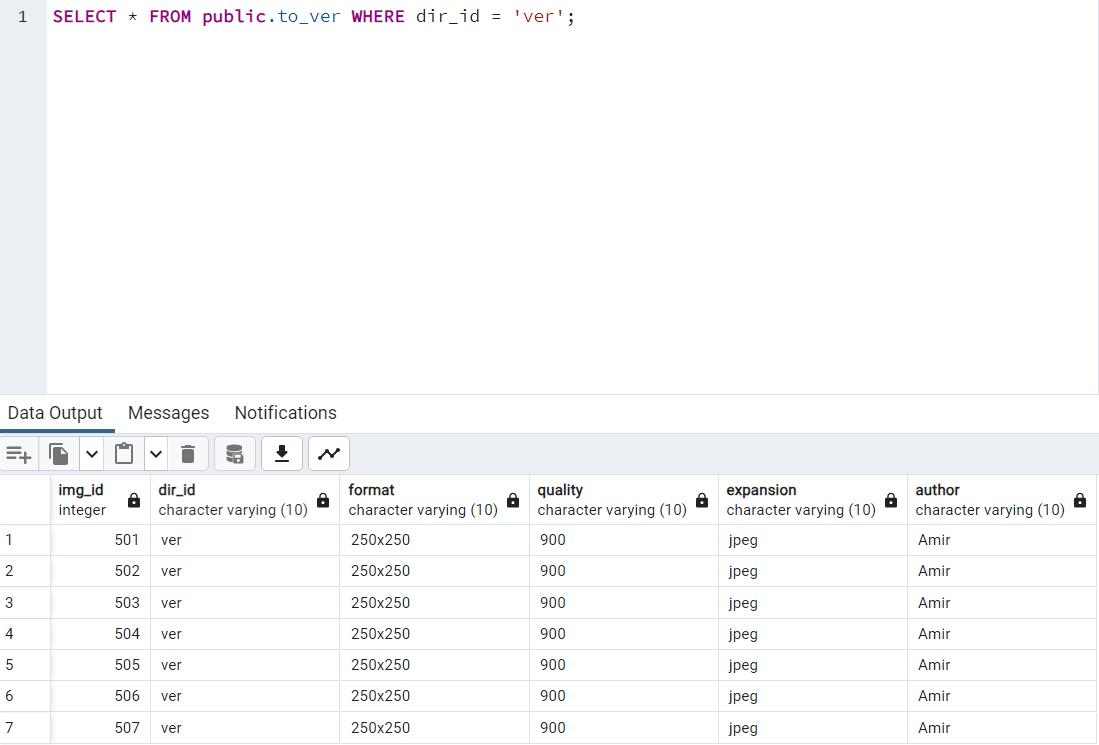


Рис 30. Пример SQL запроса

**ЗАКЛЮЧЕНИЕ**

В данной курсовой работе была разработана подсистема распознавания рисунка проводящего монтажа на послойных чертежах печатных плат с использованием нейронной сети. Целью работы было создание автоматизированной системы, способной распознавать связи проводящего монтажа на чертежах печатных плат с высокой точностью.

Для достижения поставленной цели был выбран метод градиентного спуска, основанный на использовании сверточной нейронной сети (CNN). Этот метод был выбран из-за его способности эффективно работать с нейронными сетями и извлекать признаки из них.

Результаты работы показали, что разработанная подсистема распознавания рисунка проводящего монтажа на послойных чертежах печатных плат с использованием нейронной сети не достигает высокой точности и полноты распознавания связей проводящего монтажа, но мы выяснили, что для обеспечения более высокой точности необходимо иметь больший объем выборки и лучшее качество исходных данных(изображений печатных плат).

В целом, разработанная подсистема распознавания рисунка проводящего монтажа на послойных чертежах печатных плат с помощью нейронной сети представляет собой значимый шаг в автоматизации и оптимизации процесса работы с печатными платами. Ее применение может значительно упростить и ускорить анализ и контроль качества печатных плат в различных областях, таких как электроника, авиационная и медицинская промышленности.

**СПИСОК ИСТОЧНИКОВ**

1. Viacheslav Kovalevskyi, Gleb Bochkarev. (2017) Лекция -Учимся обучать нейронные сети, от теории до практики.
2. [Andrey Sozykin.](https://www.youtube.com/@AndreySozykin) (2022) Лекция Оптимизация гиперпараметров нейросети с помощью [Keras Tuner](https://github.com/keras-team/keras-tuner)
3. [Andrey Sozykin.](https://www.youtube.com/@AndreySozykin) (2022) Лекция "Решение задачи регрессии"
4. [Andrey Sozykin.](https://www.youtube.com/@AndreySozykin) (2022) Лекция "Как подготовить собственный набор изображений для обучения нейронной сети в Keras".
5. Лонг, Дж., Шелхамер, Э., и Даррелл, Т. (2015). Полностью сверточные сети для семантической сегментации. В материалах конференции IEEE по компьютерному зрению и распознаванию образов (стр. 3431-3440).
6. Лин, Т.Ю., Доллар, П., Гиршик, Р., Хе, К., Харихаран, Б., и Белонги, С. (2017). Сети-пирамиды для обнаружения объектов. В материалах конференции IEEE по компьютерному зрению и распознаванию образов (стр. 2117–2125).
7. Хе, К., Чжан, X., Рен, С., и Сунь, Дж. (2016). Глубокое остаточное обучение для распознавания изображений. В материалах конференции IEEE по компьютерному зрению и распознаванию образов (стр. 770-778).

ПРИЛОЖЕНИЕ 1

from keras.datasets import cifar10

import xmodule

import os

from keras.models import Sequential

from keras.layers import Dense, Flatten, Activation

from keras.layers import Dropout

from keras.layers import Conv2D

from keras.layers import MaxPooling2D

from tensorflow.keras.utils import to\_categorical

from keras.optimizers import SGD

import matplotlib.pyplot as plt

from PIL import Image

from tensorflow.keras.preprocessing import image

%matplotlib inline

import importlib

import numpy as np

# Задаем seed для повторяемости результатов

np.random.seed(42)

# Путь к папке с вашими изображениями

data\_dir = "Dataset\\2"

c = 0

# Список для хранения предобработанных изображений

preprocessed\_images = []

preprocessed\_images2 = []

# Цикл по всем файлам в папке

for filename in os.listdir(data\_dir):

if filename.endswith(".jpg"): # Проверяем, что файл - изображение (можете изменить расширение)

# Загрузите изображение и предобработайте его (если это необходимо)

img\_path = os.path.join(data\_dir, filename)

img = Image.open(img\_path).resize((250, 250)) # Пример: изменяем размер до 32x32

preprocessed\_images.append(np.array(img))

preprocessed\_images2.append(np.random.randint(0, 1))

break

# # Преобразовываем список изображений в numpy массив (при необходимости)

X\_Custom\_train = np.array(preprocessed\_images)

Y\_Custom\_train = np.array(preprocessed\_images2)

# Путь к папке с вашими изображениями

data\_dir1 = "Dataset\\1"

c = 0

# Список для хранения предобработанных изображений

preprocessed\_images1 = []

preprocessed\_images12 = []v

# Цикл по всем файлам в папке

for filename in os.listdir(data\_dir1):

if filename.endswith(".jpg"): # Проверяем, что файл - изображение (можете изменить расширение)

# Загрузите изображение и предобработайте его (если это необходимо)

img\_path1 = os.path.join(data\_dir1, filename)

img1 = Image.open(img\_path1).resize((250, 250)) # Пример: изменяем размер до 32x32

preprocessed\_images1.append(np.array(img1))

preprocessed\_images12.append(np.random.randint(0, 1))

break

# # Преобразовываем список изображений в numpy массив (при необходимости)

X\_Custom\_train1 = np.array(preprocessed\_images1)

Y\_Custom\_train1 = np.array(preprocessed\_images12)

# Загружаем данные

# (X\_train, y\_train), (X\_test, y\_test) = cifar10.load\_data()

X\_train = preprocessed\_images

# Размер мини-выборки

batch\_size = 32

# Количество классов изображений

nb\_classes = 2

# Количество эпох для обучения

nb\_epoch = 15

# Размер изображений

img\_rows, img\_cols = 32, 32

# Количество каналов в изображении: RGB

img\_channels = 3

for i in range(1,40):

img\_path = f"Dataset\\train\\not marked\\image\_{i}.jpg"

img\_path2 = f"Dataset\\train\\marked\\image\_{i}.jpg"

img = image.load\_img(img\_path, target\_size=(250, 250), color\_mode = "rgb")

img2 = image.load\_img(img\_path2, target\_size=(250, 250), color\_mode = "grayscale")

# plt.imshow(img,img2)

# plt.imshow(img2)

fig, axes = plt.subplots(1, 2, figsize=(10, 5))

axes[0].imshow(img)

axes[0].set\_title('Данные без разметки')

# Отобразите второе изображение

axes[1].imshow(img2)

axes[1].set\_title('Данные с разметкой')

# Уберите оси координат

for ax in axes:

ax.axis('off')

# Покажите график с обоими изображениями

plt.show()

# Нормализуем данные

X\_train = X\_Custom\_train.astype('float32')

X\_test = X\_Custom\_train1.astype('float32')

X\_train /= 255

X\_test /= 255

# Преобразуем метки в категории

Y\_train = to\_categorical(Y\_Custom\_train, nb\_classes)

Y\_test = to\_categorical(Y\_Custom\_train1, nb\_classes)

# Создаем последовательную модель

model = Sequential()

# Первый сверточный слой

model.add(Conv2D(250, (3, 3), padding='same',

input\_shape=(250, 250, 3), activation='relu'))

# Второй сверточный слой

model.add(Conv2D(250, (3, 3), activation='relu', padding='same'))

# Первый слой подвыборки

model.add(MaxPooling2D(pool\_size=(2, 2)))

# Слой регуляризации Dropout

model.add(Dropout(0.25))

# Третий сверточный слой

model.add(Conv2D(250, (3, 3), padding='same', activation='relu'))

# Четвертый сверточный слой

model.add(Conv2D(250, (3, 3), activation='relu'))

# Второй слой подвыборки

model.add(MaxPooling2D(pool\_size=(2, 2)))

# Слой регуляризации Dropout

model.add(Dropout(0.25))

# Слой преобразования данных из 2D представления в плоское

model.add(Flatten())

# Полносвязный слой для

model.add(Dense(500, activation='relu'))

# Слой регуляризации Dropout

model.add(Dropout(0.5))

# Выходной полносвязный слой

model.add(Dense(nb\_classes, activation='softmax'))

# # Задаем параметры оптимизации

# sgd = SGD(learning\_rate=0.01, decay=1e-6, momentum=0.9, nesterov=True)

# model.compile(loss='categorical\_crossentropy',

# optimizer=sgd,

# metrics=['accuracy'])

# Задаем параметры оптимизации

# nesterov=True

sgd = SGD(learning\_rate=0.0001, momentum=0.5)

model.compile(loss='categorical\_crossentropy',

optimizer=sgd,

metrics=['accuracy'])

# Обучаем модель

history = model.fit(X\_train, Y\_train,

batch\_size=batch\_size,

epochs=nb\_epoch,

shuffle=True,

verbose=2)

plt.plot(history.history['loss'])

plt.title('Model Loss')

plt.xlabel('Epoch')

plt.ylabel('Loss')

plt.show()

# Оцениваем качество обучения модели на тестовых данных

scores = model.evaluate(X\_test, Y\_test, verbose=0)

loss = model.evaluate(X\_test, Y\_test, verbose=0)

print("Точность работы на тестовых данных: %.2f%%" % ((loss[0])\*100))

input\_file = 'Dataset\\test\\not marked'

importlib.reload(xmodule)

output\_file = 'Dataset\\test\\marked'

xmodule.predict(output\_file)

n = np.random.randint(1,7)

img\_path = f"Dataset\\test\\not marked\\image\_{n}.jpg"

img\_path2 = f"Dataset\\test\\marked\\image\_{n}.jpg"

img = image.load\_img(img\_path, target\_size=(250, 250), color\_mode = "rgb")

img2 = image.load\_img(img\_path2, target\_size=(250, 250), color\_mode = "rgb")

# plt.imshow(img,img2)

# plt.imshow(img2)

fig, axes = plt.subplots(1, 2, figsize=(10, 5))

axes[0].imshow(img)

axes[0].set\_title('Данные без разметки')

# Отобразите второе изображение

axes[1].imshow(img2)

axes[1].set\_title('РЕЗУЛЬТАТ')

# Уберите оси координат

for ax in axes:

ax.axis('off')

# Покажите график с обоими изображениями

plt.show()

import os

Rename.py

# Путь к директории с вашими файлами

dataset\_dir = 'Dataset\\train\\marked'

# Получаем список файлов в директории

files = os.listdir(dataset\_dir)

# Перебираем файлы и переименовываем их

for i, file\_name in enumerate(files):

# Генерируем новое имя файла с использованием номера

new\_file\_name = f'image\_{i + 1}.jpg' # Например, 'image\_1.jpg', 'image\_2.jpg' и так далее

# Формируем полные пути к старому и новому файлам

old\_path = os.path.join(dataset\_dir, file\_name)

new\_path = os.path.join(dataset\_dir, new\_file\_name)

# Переименовываем файл

os.rename(old\_path, new\_path)

print(f'Renamed: {file\_name} -> {new\_file\_name}')