Минобрнауки России

Юго-Западный государственный университет

Кафедра программной инженерии

ОТЧЕТ

о преддипломной (производственной) практике

наименование вида и типа практики

на (в) ООО «Предприятие ВТИ-Сервис»

наименование предприятия, организации, учреждения

Студента 4курса, группы ПО-02б

курса, группы

Тягунова Владимира Вячеславовича

фамилия, имя, отчество

Руководитель практики от предприятия, организации, учреждения

директор

должность, звание, степень

Куркина А. В.

фамилия и. о.

Оценка _____

_____

подпись, дата

Руководитель практики от университета

к.т.н. доцент

должность, звание, степень

Чаплыгин А. А.

фамилия и. о.

Оценка _____

_____

подпись, дата

Члены комиссии

_____ _____

подпись, дата фамилия и. о.

_____ _____

подпись, дата фамилия и. о.

_____ _____

подпись, дата фамилия и. о.

Курск 2024 г.

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ИНС - Искусственная нейронная система SOM(Self-Organizing Maps) - Самоорганизующиеся карты ANFIS(Adaptive Neuro Fuzzy Inference System) - Адаптивная система нейро-нечёткого вывода

1 Анализ предметной области

1.1 Понятие искусственной нейронной сети

Нейронная сеть, также именуемая искусственной нервной сетью (ИНС), это математическая модель, которая послужила программным или аппаратным воплощением принципов функционирования биологических нейронных сетей - сетей нервных клеток живых организмов. Этот термин возник в результате изучения процессов, происходящих в мозге, и попыток эмулировать их. Первые усилия по созданию нейронных сетей были предприняты У. Маккалоком и У. Питтсом. С развитием алгоритмов обучения этих моделей стали широко применяться в практических задачах, таких как предсказание, распознавание образов, управление и прочие.

Искусственная нейронная сеть (ИНС) - это архитектура, в которой простые вычислительные элементы, называемые искусственными нейронами, взаимодействуют между собой для выполнения сложных задач. Каждый нейрон обрабатывает входные сигналы и передает выходные сигналы другим нейронам в сети. Хотя отдельные нейроны могут быть простыми, их коллективное взаимодействие в большой сети позволяет выполнять сложные вычисления и задачи.

Искусственная нейронная сеть (ИНС) представляет собой комплекс систематически связанных и взаимодействующих между собой простых вычислительных элементов, называемых искусственными нейронами. В контексте машинного обучения, они функционируют как специализированный вид методов для распознавания образов и дискриминантного анализа. Математически они представляют собой сложную многопараметрическую задачу нелинейной оптимизации. В кибернетике они используются для адаптивного управления и в робототехнике. С точки зрения развития вычислительной техники и программирования, они являются мощным инструментом для решения проблемы эффективного параллелизма.

Важно отметить, что нейронные сети не программированны в традиционном понимании этого термина; они обучаются. Процесс обучения заклю-

чается в настройке параметров связей между нейронами на основе предоставленных данных. Этот процесс позволяет нейронным сетям обнаруживать сложные закономерности в данных, выполнять обобщение и возвращать верные результаты даже на основе данных, которые не были представлены в процессе обучения или были представлены с искажениями.

1.1.1 Распознавание образов и классификация

Когда мы говорим о различных ”образах мы имеем в виду разнообразные объекты, такие как текстовые символы, изображения, звуковые образцы и так далее. В процессе обучения нейронной сети предъявляются эти различные образы, каждому из которых присваивается определенный класс. Образец представляется в виде вектора значений признаков, и сеть учится определять, к какому классу относится каждый образец. Важно, чтобы набор признаков однозначно определял класс образца. Если признаков недостаточно, сеть может неправильно классифицировать образец, связывая его с несколькими классами.

После завершения обучения количество нейронов в выходном слое сети обычно соответствует количеству классов. Каждый нейрон в выходном слое представляет определенный класс, и сеть выдаёт ответ о принадлежности образца к тому или иному классу. Если сеть уверена в классификации, на одном из выходов появится признак принадлежности к классу, а на других выходах этот признак отсутствует. Однако, если сеть не уверена, может возникнуть ситуация, когда на нескольких выходах присутствует признак принадлежности к классу, что указывает на неопределённость сети в своём ответе.

1.1.2 Образы и Классификация

При обучении нейронной сети различные типы данных, такие как текст, изображения и звук, представлены в виде образов и привязаны к определенным классам. Сеть изучает эти образы и их признаки, чтобы точно классифицировать их. Важно, чтобы эти признаки явно указывали на класс об-

разца. По завершении обучения количество нейронов в выходном слое сети соответствует количеству классов. Каждый нейрон представляет определенный класс, и сеть выдает ответ о принадлежности образца к определенному классу. В случае неопределенности сеть может указать на несколько возможных классов.

Классификация по формату входной информации:

1. Аналоговые нейронные сети: работают с информацией в форме действительных чисел.

2. Двоичные нейронные сети: оперируют с информацией, представленной в двоичном виде.

3. Образные нейронные сети: оперируют с информацией, представленной в виде образов, таких как знаки, иероглифы или символы.

Классификация по типу обучения:

1. Обучение с учителем: Нейронная сеть использует известные пары входных данных и соответствующих им выходных значений для обучения. В этом случае выходное пространство решений известно заранее.

2. Обучение без учителя: Нейронная сеть формирует выходное пространство решений только на основе входных данных, без предоставления соответствующих выходных значений. Такие сети называют самоорганизующимися, так как они обнаруживают структуры или паттерны в данных без явного учителя.

3. Обучение с подкреплением: В этом типе обучения система взаимодействует со средой, принимая последовательность действий и получая за них награды или наказания. Цель состоит в том, чтобы оптимизировать стратегию действий так, чтобы максимизировать суммарную награду в долгосрочной перспективе.

Существует два типа классификации синапсов по характеру настройки:

1. Сети с фиксированными связями: Весовые коэффициенты нейронной сети выбираются заранее и остаются неизменными на протяжении работы сети. Это означает, что они не подвергаются изменениям в процессе обучения и задаются исходными условиями задачи.

2. Сети с динамическими связями: В этих сетях весовые коэффициенты синапсов настраиваются в процессе обучения. Это позволяет сети адаптироваться к новой информации и улучшать свою производительность в зависимости от задачи или окружающей среды.

3. Обучение с подкреплением: В этом типе обучения система взаимодействует со средой, принимая последовательность действий и получая за них награды или наказания. Цель состоит в том, чтобы оптимизировать стратегию действий так, чтобы максимизировать суммарную награду в долгосрочной перспективе.

Классификация по характеру связей в нейронных сетях

1. Нейронные сети прямого распространения:

Все связи направлены строго от входных нейронов к выходным.

Примеры: перцептрон Розенблатта, многослойный перцептрон, сети Ворда.

2. Рекуррентные нейронные сети:

Сигнал с выходных нейронов или нейронов скрытого слоя частично передаётся обратно на входы нейронов входного слоя (обратная связь).

Рекуррентная сеть Хопфилда решает задачи компрессии данных и построения ассоциативной памяти.

Частный случай: двунаправленные сети.

3. Радиально-базисные функции (RBF):

Используются нейронные сети с единственным скрытым слоем.

Нелинейная активационная функция только у нейронов скрытого слоя.

Синаптические веса связей входного и скрытого слоёв равны единице.

4. Самоорганизующие карты (Self-Organizing Maps, SOM), представляют собой модель нейронных сетей, которая используется для визуализации и кластеризации данных. Это метод проецирования данных из многомерного пространства в пространство с более низкой размерностью, обычно двумерное. SOM также применяются в моделировании, прогнозировании и других задачах. Они являются разновидностью сетей Кохонена.

В сети Кохонена сигнал поступает на все нейроны одновременно, а выходной сигнал формируется по принципу ”победитель забирает всё”. В процессе обучения веса синапсов настраиваются таким образом, чтобы узлы сети описывали кластерную структуру данных. Удобно представлять SOM как двумерную сетку узлов в многомерном пространстве.

Начальное вложение сетки в пространство данных выбирается произвольно, а затем узлы сети перемещаются на каждом этапе обучения в направлении данных. Алгоритм обучения состоит из двух этапов: грубой настройки, где узлы двигаются коллективно для грубого отображения структуры данных, и тонкой настройки, где настраиваются индивидуальные положения узлов.

Этот процесс повторяется определённое число эпох, причем количество шагов может изменяться в зависимости от задачи.

1.1.3 Кластеризация и Новые Классы

Кластеризация подразумевает разделение входных сигналов на классы, неизвестные заранее по числу и признакам. После обучения сеть может определить, к какому классу относится входной сигнал, либо указать на его новизну. Такие сети могут обнаруживать новые, ранее неизвестные классы сигналов. Соответствие между выделенными сетью классами и классами в предметной области устанавливается человеком.

1.1.4 Архитектура Нейронных Сетей

Нейронные сети Кохонена имеют ограниченный размер, разделяясь на гиперслои и ядра. Идеальное количество параллельных слоев ограничено до 112, где каждый слой содержит от 500 до 2000 микроколонок. Эти микроколонки обеспечивают кодирование и вывод результатов. Регулирование числа нейронов и слоев осуществляется с помощью суперкомпьютеров, делая нейронные сети пластичными и адаптивными.

1.1.5 Прогнозирование

Способность нейронной сети к прогнозированию происходит из ее способности к обобщению и выявлению скрытых зависимостей между входными и выходными данными. После обучения сеть может предсказать будущее значение последовательности, основываясь на предыдущих значениях и/или текущих факторах. Прогнозирование возможно лишь в случае, если предыдущие изменения в некоторой степени влияют на будущие. Например, прогнозирование цен акций на основе предыдущих недельных котировок может быть успешным (но не обязательно), в то время как прогнозирование результатов лотереи на основе 50-летних данных практически бесполезно.

1.1.6 Аппроксимация

Способность нейронной сети к прогнозированию напрямую вытекает из ее способности обобщать и выявлять скрытые зависимости между входными и выходными данными. После обучения сеть может предсказывать будущее значение последовательности, опираясь на предыдущие значения и/или текущие факторы. Прогнозирование возможно только в том случае, если предыдущие изменения в некоторой степени определяют будущие. Например, прогнозирование цен акций на основе предыдущих недельных котировок может быть успешным (но не обязательно), в то время как прогнозирование результатов лотереи на основе 50-летних данных практически бесполезно.

Нейронные сети способны аппроксимировать непрерывные функции. Обобщённая аппроксимационная теорема показывает, что с помощью линейных операций и каскадного соединения можно получить устройство, способное вычислить любую непрерывную функцию с некоторой наперёд заданной точностью. Это означает, что нейроны могут иметь различные нелинейные характеристики, от сигмоидальной до волновых пакетов или вейвлетов, синусов или многочленов. Выбор нелинейной функции может влиять на сложность сети, но при правильном выборе структуры нейронная сеть остаётся

универсальным аппроксиматором и может достаточно точно аппроксимировать функционирование любого непрерывного автомата.

1.1.7 Сжатие данных и ассоциативная память

Нейросети обладают способностью выявлять взаимосвязи между различными параметрами, что позволяет более компактно представлять данные большой размерности в случае их тесной взаимосвязи. Процесс обратного восстановления исходного набора данных из части информации называется (авто)ассоциативной памятью. Ассоциативная память также способна восстанавливать исходный сигнал или образ из зашумленных или повреждённых входных данных. Решение задачи гетероассоциативной памяти позволяет реализовать память, адресуемую по содержимому.

1.2 Понятие нечёткой логики

Нечеткая логика представляет собой раздел математики, который расширяет традиционную логику и теорию множеств, используя концепцию нечетких множеств. Она была впервые предложена Лотфи Заде в 1965 году. В отличие от классической логики, где элементы либо принадлежат множеству (имеют значение 1), либо не принадлежат (имеют значение 0), нечеткие множества могут иметь значения на интервале [0, 1], отражая степень принадлежности элемента к множеству. На основе этой концепции разрабатываются различные логические операции и определяются лингвистические переменные, значениями которых являются нечеткие множества.

Область применения нечеткой логики включает исследование рассуждений в условиях нечеткости, размытости, аналогичных рассуждениям в обычной логике, а также их применение в вычислительных системах.

1.2.1 Символическая нечёткая логика

Нечёткая логика, также известная как символическая нечёткая логика, базируется на концепции t-нормы. После выбора определённой t-нормы появляется возможность определить основные операции над пропозициональ-

ными переменными: конъюнкцию, дизъюнкцию, импликацию, отрицание и другие.

Теорема о дистрибутивности, свойственная классической логике, выполняется лишь при использовании t-нормы Гёделя. Импликация обычно определяется операцией, называемой residium, которая, в свою очередь, зависит от выбранной t-нормы.

Эти базовые операции приводят к формальному определению базовой нечёткой логики, имеющей сходства с классической булевой логикой (исчислением высказываний).

Существуют три основные базовые нечёткие логики: логика Лукасевича, логика Гёделя и вероятностная логика. Интересно, что объединение любых двух из этих логик приводит к классической булевой логике.

1.2.2 Синтез функций непрерывной логики заданных таблично

Функция нечёткой логики Заде всегда принимает значение одного из своих аргументов либо его отрицания. Таким образом, функцию нечёткой логики можно задать таблицей выбора, в которой перечислены все варианты упорядочения аргументов и отрицаний, и для каждого варианта указано значение функции.

Однако не любая произвольная таблица выбора задаёт функцию нечёткой логики. В одной работе был сформулирован критерий, позволяющий определить, является ли функция, заданная таблицей выбора, функцией нечёткой логики, и предложен простой алгоритм синтеза, основанный на концепциях конституента минимума и максимума. Функция нечёткой логики представляет собой дизъюнкцию конституент минимума, где конституента максимума — это конъюнкция переменных текущей области, больших либо равных значению функции в этой области (справа от значения функции в неравенстве, включая значение функции).

1.2.3 Теория приближённых вычислений

В общем, основное понятие нечёткой логики можно описать как использование нечётких множеств, которые определяются через обобщённую характеристическую функцию. Это позволяет работать с нечёткими отношениями, объединениями, пересечениями и дополнениями множеств. Важным элементом является лингвистическая переменная.

Для применения нечёткой логики в некоторых сферах достаточно этого минимального набора определений. Однако для большинства случаев требуется также определить правила вывода и оператор импликации.

1.2.4 Нечёткая логика и нейронные сети

Основная идея нечёткой логики в широком смысле заключается в использовании нечётких множеств, которые определяются через обобщённую характеристическую функцию. Путем введения операций объединения, пересечения и дополнения множеств (через характеристическую функцию) а также концепции нечётких отношений и лингвистических переменных, создается база для приложения нечёткой логики в различных областях.

Важно отметить, что для некоторых применений этого подхода достаточно указанных минимальных определений. Однако для большинства случаев необходимо определить правила вывода и оператор импликации.

Кроме того, поскольку нечёткие множества могут быть представлены функциями принадлежности, а t-нормы и k-нормы являются обычными математическими операциями, возможно представление нечётких логических рассуждений в виде нейронной сети. Здесь функции принадлежности интерпретируются как функции активации нейронов, передача сигналов как связи, а логические t-нормы и k-нормы как специальные виды нейронов, выполняющие соответствующие математические операции. Существует разнообразие подобных нейро-нечётких сетей, включая ANFIS (Adaptive Neuro Fuzzy Inference System) - адаптивную нейро-нечеткую систему вывода. О ней чуть расскажу чуть позже.

1.2.5 Байесовская вероятность

Связь между нечёткой логикой и байесовской вероятностью выражается через байесовскую логико-вероятностную модель нечёткого вывода. Эта модель трансформирует нечёткие продукции в вероятностные функции, определяющие апостериорное распределение на множестве гипотез, соответствующих значениям выходной лингвистической переменной. После этого происходит дефаззификация для получения чёткого значения выходной переменной.

Неонечтокие продукции состоят из правил типа ”ЕСЛИ ..., ТО ... где посылка и заключение содержат лингвистические переменные, а лингвистические переменные имеют терм-множества с функциями принадлежности.

Байесовская логико-вероятностная модель нечёткого вывода рассматривает значения выходной переменной как гипотезы и определяет их априорные вероятности. Новая информация поступает в виде значений входных переменных, которые служат свидетельствами в пользу или против гипотез.

Условные вероятности заключений при данных посылках определяются через функции принадлежности нечётких множеств.

Используя теорему Байеса, априорные вероятности обновляются с учетом новой информации, что позволяет получить апостериорное распределение вероятностей на множестве гипотез.

Для дефаззификации можно использовать различные методы, такие как метод максимума апостериорной вероятности (MAP) или метод среднего значения апостериорной вероятности (MEP).

1.3 Понятие Адаптивной системы нейро-нечеткого вывода

Адаптивная система нейро-нечеткого вывода (ANFIS) представляет собой разновидность искусственной нейронной сети, основанной на системе нечеткого вывода Такаги-Сугено. Разработанная в начале 1990-х годов, эта методика объединяет в себе нейронные сети и принципы нечеткой логики,

что дает ей потенциал для использования преимуществ обоих в одной структуре.

В ANFIS система вывода соответствует набору нечетких правил ”ЕСЛИ–ТО способных к обучению для аппроксимации нелинейных функций. Следовательно, ANFIS считается универсальным оценщиком. Для более эффективного и оптимального использования ANFIS можно воспользоваться наилучшими параметрами, полученными с использованием генетического алгоритма. Эта технология находит применение в интеллектуальных системах управления энергопотреблением.

ANFIS представляет собой разновидность искусственной нейронной сети, базирующейся на нечеткой системе вывода Такаги-Сугено. Разработанная в начале 1990-х годов, эта методика объединяет в себе нейронные сети и принципы нечеткой логики, что дает ей потенциал для использования преимуществ обоих в одной структуре.

Архитектура ANFIS В структуре сети можно выделить две части: исходную и последующую. Архитектура состоит из пяти уровней:

1. Уровень фаззификации: Принимает входные значения и определяет функции принадлежности, принадлежащие им. Степени принадлежности вычисляются на основе исходных параметров.

2. Уровень правил: Генерирует сильные стороны для правил на основе вторичных параметров.

3. Уровень нормализации: Нормализует вычисленную силу срабатывания путем деления каждого значения на общую силу срабатывания.

4. Уровень следствия: Принимает нормализованные значения и параметры следствия. Возвращает дефаззифицированные значения.

5. Выходной уровень: Получает дефаззифицированные значения и возвращает конечный результат.

ANFIS позволяет аппроксимировать нелинейные функции, делая его универсальным оценщиком с использованием генетического алгоритма для оптимизации его параметров. Эта технология находит применение в интеллектуальных системах управления энергопотреблением.

1.3.1 Слой фаззификации

Первый уровень в сети ANFIS представляет собой ключевое отличие от обычной нейронной сети. Обычно нейронные сети работают с этапом предварительной обработки данных, на котором признаки преобразуются в нормализованные значения от 0 до 1. Однако в ANFIS нет необходимости в использовании сигмоидальной функции. Вместо этого происходит преобразование числовых значений в нечеткие.

Давайте рассмотрим пример: предположим, у нас есть сеть, которая получает на вход расстояние между двумя точками в 2D-пространстве. Это расстояние измеряется в пикселях и может принимать значения от 0 до 500 пикселей. Преобразование числовых значений в нечеткие выполняется с использованием функций принадлежности, которые состоят из семантических описаний, таких как ”близко ”средне”и ”дальше”. Каждое из этих лингвистических значений соответствует отдельному нейрону. Например, нейрон ”близко”срабатывает с некоторым значением от 0 до 1, если расстояние попадает в категорию ”близко”. Точно так же нейрон ”средне”срабатывает, если расстояние соответствует этой категории. Таким образом, входное значение ”расстояние в пикселях”разбивается на три разных нейрона для каждой категории: ”близко ”средне”и ”дальше”.

2 Техническое задание

2.1 Основание для разработки

Основанием для разработки является задание на выпускную квалификационную работу бакалавра «Интеллектуальная система распознавания объектов по цветовым характеристикам на основе нечетких нейронных сетей».

2.2 Цель и назначение разработки

Основной задачей выпускной квалификационной работы является разработка интеллектуальной системы распознавания объектов на основе их цветовых характеристик с использованием нечетких нейронных сетей. Система должна обеспечивать высокую точность распознавания объектов различных классов на основе их цветовых параметров.

Задачами данной разработки являются:

- сбор набора данных, содержащего изображения объектов различных классов;
- предварительная обработка изображений для выделения цветовых характеристик объектов;
- проектирование архитектуры нечеткой нейронной сети;
- обучение нейронной сети на подготовленных данных;
- создание удобного поиска по сайту;
- оптимизация параметров сети для достижения максимальной точности распознавания;
- создание программного интерфейса для взаимодействия с разработанной нейронной сетью.

2.3 Требования пользователя к интерфейсу приложения

Приложение должно включать в себя:

- графический интерфейс пользователя;
- возможность загрузки изображений объектов для их распознавания;

- отображение результатов распознавания с указанием класса объекта и уверенности в распознавании;

- возможность обучения системы на пользовательских данных для улучшения качества распознавания;

- интерфейс для добавления новых классов объектов и их обучения;

- возможность экспорта результатов распознавания в формате, подходящем для дальнейшей обработки или анализа

Композиция шаблона приложения представлена на рисунке 2.1.

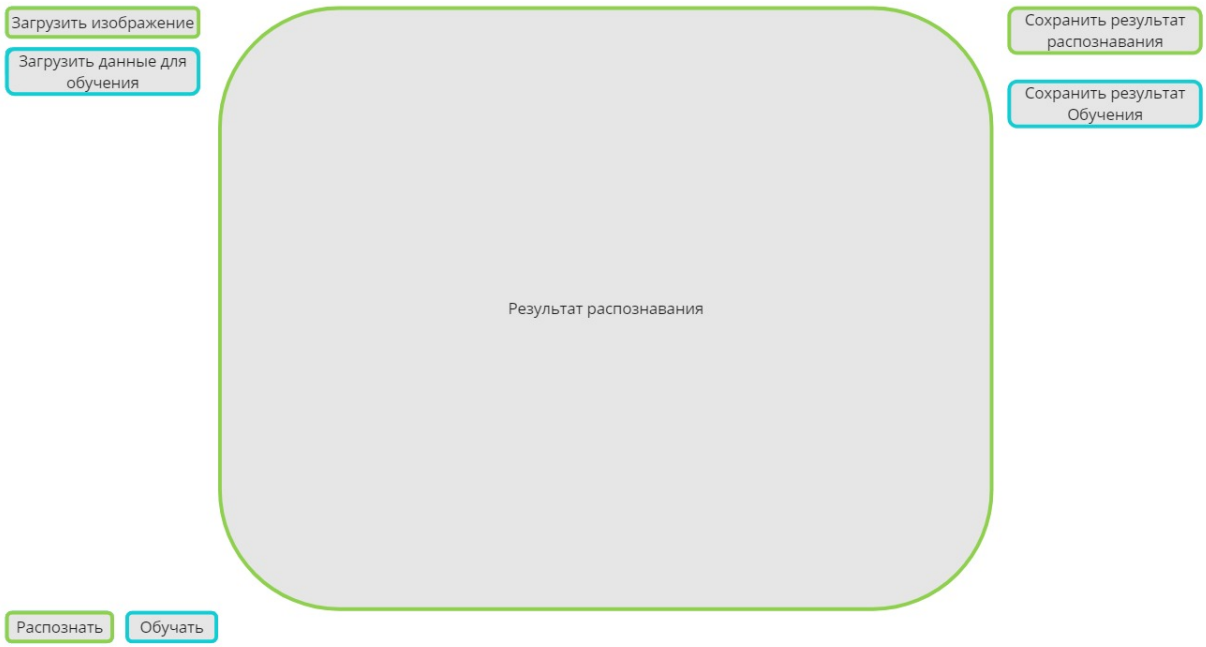

Рисунок 2.1 – Композиция шаблона приложения

2.4 Моделирование вариантов использования

Для моделирования вариантов использования разрабатываемого приложения была использована диаграмма вариантов использования, представленная на рисунке 2.2.

На диаграмме представлены основные варианты использования приложения, включая загрузку изображений для распознавания, обучение системы, добавление новых классов объектов и экспорт результатов распознава-

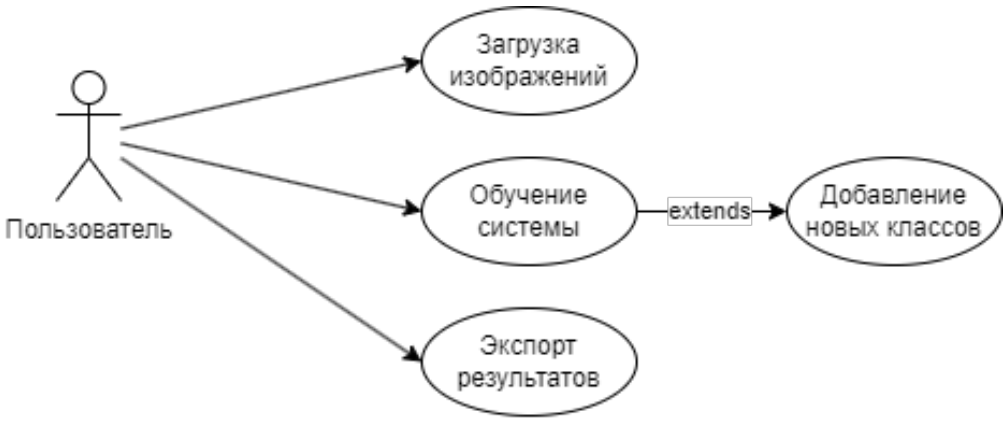

Рисунок 2.2 – Диаграмма вариантов использования

ния. Данная диаграмма помогает понять основные функциональные возможности приложения и взаимодействие с пользователями.

2.5 Требования к оформлению документации

Разработка программной документации и программного изделия должна производиться согласно ГОСТ 19.102-77 и ГОСТ 34.601-90. Единая система программной документации.

3 Технический проект

3.1 Общая характеристика организации решения задачи

Этот проект направлен на разработку системы, которая использует нечеткие нейронные сети для распознавания объектов на основе цветовых характеристик. Система будет способна анализировать изображения и выделять объекты, соответствующие заданным цветовым параметрам.

Основная цель - создание эффективной и точной системы распознавания объектов. Задачи включают:

- Разработка алгоритма нечеткой нейронной сети;
- Создание базы данных для обучения и тестирования системы;
- Интеграция системы с камерами для реального времени обработки изображений.

3.2 Обоснование выбора технологии проектирования

Нечеткие нейронные сети сочетают принципы нечеткой логики и нейронных сетей, что позволяет системе обрабатывать нечеткие и неточные данные, характерные для реальных изображений.

3.2.1 Описание используемых технологий и языков программирования

В процессе разработки приложения используются программные средства и языки программирования. Каждое программное средство и каждый язык программирования применяется для круга задач, при решении которых они необходимы.

3.2.2 Нечеткие нейронные сети

Комбинация нечеткой логики и нейронных сетей позволяет системе обрабатывать нечеткие данные. Мы используем нечеткие нейронные сети для улучшения точности распознавания объектов по цветовым характеристикам, даже когда данные имеют шум или неточности.

3.2.3 Машинное обучение

Применение алгоритмов машинного обучения для анализа и классификации данных. В нашем случае, мы обучаем модель на большом наборе изображений, чтобы система могла эффективно распознавать и классифицировать объекты в реальном времени.

3.2.4 Python и его библиотеки

Python является предпочтительным языком программирования благодаря своей читаемости, простоте и обширной экосистеме библиотек, подходящих для работы с данными и машинным обучением:

- NumPy: Используется для эффективной работы с массивами и матрицами, что критично для обработки изображений и численных вычислений.
- Pandas: Предоставляет удобные структуры данных для анализа и манипуляции данными.
- Matplotlib/Seaborn: Библиотеки для визуализации данных, которые помогают в анализе результатов и представлении данных.
- OpenCV: Открытая библиотека для работы с компьютерным зрением, которая может использоваться для предварительной обработки изображений.
- TensorFlow/Keras: Популярные фреймворки для глубокого обучения, которые предоставляют инструменты для создания, обучения и тестирования нейронных сетей.
- Scikit-learn: Библиотека для машинного обучения, предоставляющая различные алгоритмы классификации, регрессии и кластеризации.

3.2.4.1 Достоинства языка Python

Использование Python и его библиотек для создания системы распознавания объектов имеет ряд преимуществ:

- Простота и читаемость: Python известен своим чистым и легко читаемым синтаксисом, что упрощает написание и поддержку кода. Это особенно важно в больших проектах и при работе в команде.

- Богатая экосистема: Python обладает обширной экосистемой библиотек и фреймворков, которые значительно ускоряют разработку и предоставляют готовые решения для многих задач.

- Поддержка сообщества: Огромное сообщество разработчиков Python постоянно обновляет и улучшает существующие библиотеки, а также создает новые, что делает язык всегда актуальным.

- Гибкость: Python позволяет интегрировать различные технологии и подходы, такие как нечеткая логика и нейронные сети, что идеально подходит для нашей системы.

- Мультипарадигменность: Python поддерживает различные стили программирования — объектно-ориентированный, процедурный и функциональный, что дает гибкость в выборе подхода к решению задач.

- Открытый исходный код: Большинство библиотек Python доступны бесплатно и имеют открытый исходный код, что снижает затраты на разработку.

- Интеграция с другими языками: Python может быть легко интегрирован с другими языками программирования, что позволяет использовать его в качестве ”клея”для различных компонентов системы.

- Масштабируемость: Python и его библиотеки подходят как для прототипирования, так и для масштабирования систем на производство.

- Поддержка научных вычислений: Библиотеки, такие как NumPy и SciPy, предоставляют мощные инструменты для научных вычислений, что необходимо при обработке изображений и данных.

- Визуализация данных: С помощью Matplotlib и Seaborn можно легко создавать графики и визуализации, что полезно для анализа данных и результатов системы.

- Машинное обучение и глубокое обучение: TensorFlow, Keras и Scikit-learn предоставляют обширные возможности для создания, обучения и тестирования моделей машинного обучения и глубокого обучения.

Эти преимущества делают Python и его библиотеки идеальным выбором для разработки системы распознавания объектов, особенно когда требуется работа с большими объемами данных и сложными алгоритмами.

3.2.4.2 Недостатки языка Python

- Несмотря на множество преимуществ Python, существуют и недостатки, которые следует учитывать при выборе языка для проекта:

- Скорость выполнения: Python часто критикуют за его относительно медленную скорость выполнения по сравнению с компилируемыми языками, такими как C или C++. Это связано с его динамической типизацией и интерпретируемостью.

- Потребление памяти: Программы на Python могут потреблять больше памяти, что может быть проблемой для систем с ограниченными ресурсами.

- Мобильная разработка: Python не является идеальным выбором для мобильной разработки. Другие языки, такие как Swift и Kotlin, лучше подходят для создания мобильных приложений.

- Встроенные ограничения: Python имеет некоторые встроенные ограничения, такие как Global Interpreter Lock (GIL), который может затруднять полноценное использование многоядерных процессоров в многопоточных программах.

- Игровая разработка: Хотя существуют библиотеки для создания игр на Python, он не является основным выбором для профессиональной разработки игр, где предпочтение отдают таким языкам, как C++ или C# .

- Строгость типизации: Python использует динамическую типизацию, что может привести к ошибкам во время выполнения, которые были бы обнаружены на этапе компиляции в языках со строгой типизацией.

- Базовые настройки и оптимизация: Для достижения максимальной производительности может потребоваться дополнительная работа по оптимизации и настройке, включая использование сторонних инструментов.

- Зависимости: Управление зависимостями в Python может быть сложным, особенно когда проекты становятся большими и сложными.

- Безопасность: Python имеет открытый исходный код, что может представлять определенные риски безопасности, если не принимать соответствующие меры.

- Работа с базами данных: Взаимодействие с некоторыми базами данных может быть менее эффективным по сравнению с языками, специализированными на этом, например, SQL.

Эти недостатки не делают Python плохим выбором; скорее, они подчеркивают важность тщательного анализа требований проекта и возможных ограничений перед выбором инструментов для его реализации.

3.3 Диаграмма компонентов

Диаграмма компонентов представляет структуру системы в виде набора компонентов и их взаимосвязей. Каждый компонент отвечает за определенную функцию в рамках системы и может включать в себя подсистемы или модули.

3.3.1 Структура компонентов

На диаграмме компонентов изображены основные блоки системы, такие как:

- Графический интерфейс пользователя: Модуль, отвечающий за взаимодействие с пользователем, представление результатов и получение входных данных.

- Модуль предварительной обработки данных: Отвечает за подготовку данных к анализу, включая фильтрацию шума и нормализацию изображений.

- Модуль нечеткой нейронной сети: Ядро системы, реализующее алгоритмы обучения и распознавания объектов.

- База данных: Хранит обучающий и тестовый наборы данных, а также результаты работы системы.

- Модуль анализа данных: Производит анализ данных, классификацию и предоставляет статистику по результатам.

3.3.2 Взаимодействие компонентов

Компоненты системы взаимодействуют друг с другом следующим образом:

1. Пользователь загружает изображение через графический интерфейс пользователя.

2. Интерфейс передает изображение в модуль предварительной обработки данных.

3. После обработки данные передаются в модуль нечеткой нейронной сети для распознавания объектов.

4. Результаты распознавания сохраняются в базе данных.

5. Модуль анализа данных извлекает результаты из базы данных и представляет их пользователю через графический интерфейс пользователя.

Диаграмма компонентов представленна на рисунке 3.1.

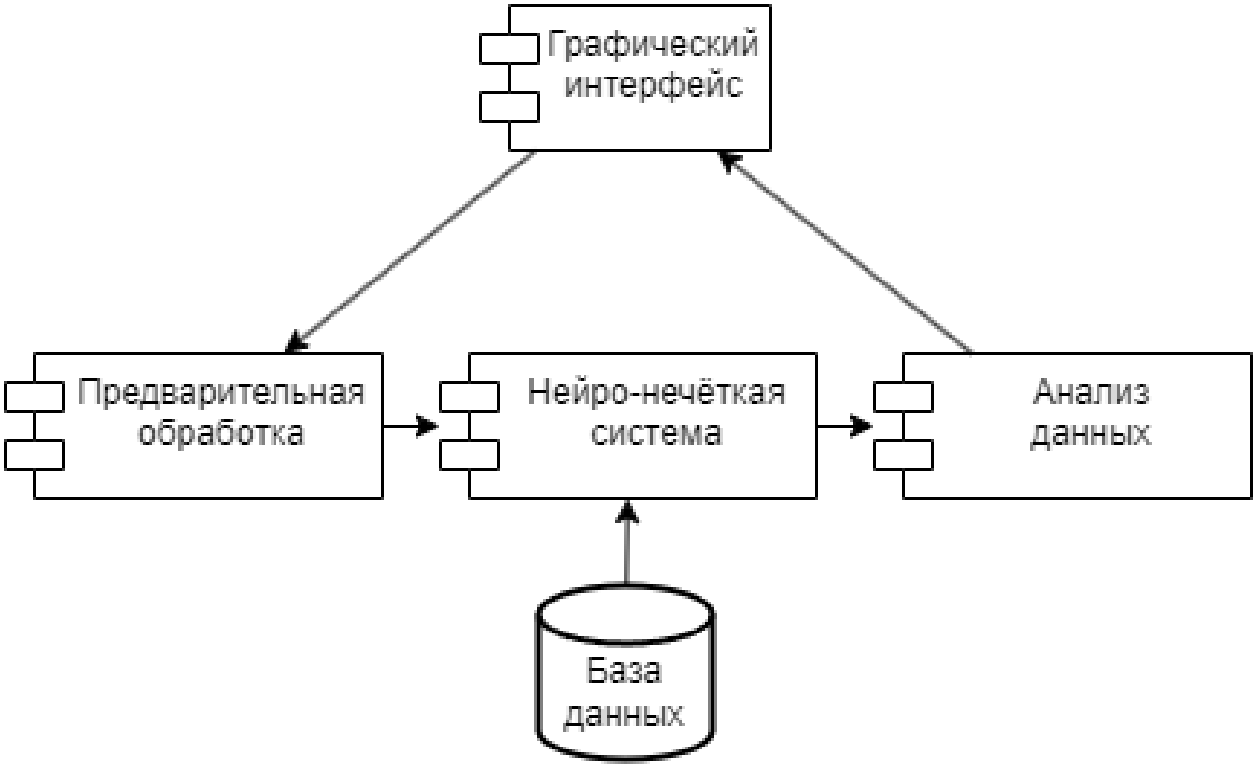

Рисунок 3.1 – Диаграмма компонентов системы

3.4 Содержание информационных блоков. Основные сущности

Информационные блоки системы распознавания объектов представляют собой структурированные данные, которые используются для обучения, тестирования и работы системы. Основные сущности, которые являются частью этих блоков, включают:

- Изображение: Основной объект анализа, содержащий объекты для распознавания.

- Объект: Целевая сущность на изображении, которую необходимо распознать и классифицировать.

- Цветовой параметр: Характеристика объекта, используемая для его идентификации и классификации.

- Метка класса: Определение категории, к которой принадлежит объект.

- Набор данных: Коллекция изображений и соответствующих меток, используемых для обучения и тестирования системы.

Каждый информационный блок содержит данные, необходимые для выполнения специфических задач в рамках системы, таких как обучение нечеткой нейронной сети или классификация и анализ объектов. Структура информационных блоков разработана таким образом, чтобы обеспечить максимальную эффективность и точность при минимальных затратах времени на обработку данных.

3.4.1 Структура информационного блока

Каждый информационный блок состоит из следующих элементов:

1. Идентификатор изображения: Уникальный идентификатор, присваиваемый каждому изображению в базе данных.

2. Идентификатор объекта: Уникальный идентификатор, присваиваемый каждому распознанному объекту.

3. Цветовые характеристики: Набор параметров, описывающих цвет объекта в различных цветовых пространствах.

4. Геометрические параметры: Размеры и форма объекта, а также его положение на изображении.

5. Классификационные метки: Метки, указывающие на принадлежность объекта к определенной категории или классу.

Эта структура позволяет системе быстро обрабатывать большие объемы данных и эффективно распознавать объекты с высокой степенью точности.

3.4.2 Взаимодействие информационных блоков с компонентами системы

Информационные блоки взаимодействуют с различными компонентами системы следующим образом:

1. Модуль предварительной обработки данных: Извлекает необходимые параметры из изображений и формирует информационные блоки.

2. Модуль нечеткой нейронной сети: Использует информационные блоки для обучения и классификации объектов.

3. База данных: Хранит информационные блоки и обеспечивает их доступность для других компонентов системы.

4. Модуль анализа данных: Производит дополнительный анализ информационных блоков для генерации отчетов и статистики.

Таким образом, информационные блоки являются ключевым элементом системы, обеспечивающим ее функционирование и достижение поставленных целей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лутц, М. Изучаем Python, 5-е издание / М. Лутц. – Санкт-Петербург : Питер, 2019. – 1584 с. – ISBN 978-5-4461-0705-9. – Текст : непосредственный.

2. Гудфеллоу, И., Бенджио, Ю., Курвилль, А. Глубокое обучение / И. Гудфеллоу, Ю. Бенджио, А. Курвилль. – Москва : ДМК Пресс, 2017. – 652 с. – ISBN 978-5-97060-487-9. – Текст : непосредственный.

3. Хайкин, С. Нейронные сети: полный курс, 2-е издание / С. Хайкин. – Москва : Вильямс, 2018. – 1104 с. – ISBN 978-5-8459-2101-0. – Текст : непосредственный.

4. Росс, Т. Дж. Нечеткая логика с приложениями в инженерных науках / Т. Дж. Росс. – Москва : Мир, 2016. – 800 с. – ISBN 978-5-03-004474-5. – Текст : непосредственный.

5. Мерфи, К. Машинное обучение: вероятностный подход / К. Мерфи. – Москва : ДМК Пресс, 2018. – 704 с. – ISBN 978-5-97060-212-7. – Текст : непосредственный.

6. Рашка, С., Мирджалили, В. Python и машинное обучение / С. Рашка, В. Мирджалили. – Москва : ДМК Пресс, 2018. – 418 с. – ISBN 978-5-97060-310-0. – Текст : непосредственный.

7. Жолковский, Е. К. TensorFlow для профессионалов / Е. К. Жолковский. – Москва : ДМК Пресс, 2019. – 480 с. – ISBN 978-5-97060-746-7. – Текст : непосредственный.

8. Чоллет, Ф. Глубокое обучение на Python / Ф. Чоллет. – Москва : ДМК Пресс, 2018. – 304 с. – ISBN 978-5-97060-409-1. – Текст : непосредственный.

9. Клейн, Р. Нечеткие системы в Python / Р. Клейн. – Москва : ДМК Пресс, 2020. – 320 с. – ISBN 978-5-97060-758-0. – Текст : непосредственный.

10. Бейдер, Д. Python Tricks: A Buffet of Awesome Python Features / Д. Бейдер. – Москва : ДМК Пресс, 2021. – 300 с. – ISBN 978-5-97060-999-7. – Текст : непосредственный.

11. Герон, О. Практическое машинное обучение с Scikit-Learn и TensorFlow / О. Герон. – Москва : ДМК Пресс, 2019. – 572 с. – ISBN 978-5-97060-524-1. – Текст : непосредственный.

12. Нильсен, М. Нейронные сети и глубокое обучение / М. Нильсен. – Москва : ДМК Пресс, 2021. – 250 с. – ISBN 978-5-97060-777-1. – Текст : непосредственный.

13. Джеймс, Д. Нечеткие системы и Python: практическое руководство / Д. Джеймс. – Москва : ДМК Пресс, 2022. – 340 с. – ISBN 978-5-97060-888-4. – Текст : непосредственный.