

Министерство образования Республики Беларусь

Учреждение образования  
Белорусский государственный университет  
информатики и радиоэлектроники

УДК 004.855.5+004.932.2

Венгеренко  
Вадим Владимирович

Алгоритм и программное средство анализа графических объектов

**АВТОРЕФЕРАТ**

на соискание степени магистра

по специальности 1-40 80 04 – Информатика и технологии программирования

Научный руководитель  
Боброва Н.Л.  
к.т.н., доцент

Минск 2024

## КРАТКОЕ ВВЕДЕНИЕ

Анализ графических объектов находит свое применение во многих отраслях: здравоохранение, транспорт, производство, сельское хозяйство, безопасность.

В настоящее время в области анализа графических объектов наибольший интерес представляют алгоритмы, основанные на методах машинного и глубокого обучения [1-А]. С анализом изображений связана такая научная область, как компьютерное зрение.

Поскольку анализ графических объектов подразумевает достаточно широкий круг задач, то в пределах данного исследования рассматривается задача классификации изображений.

Среди множества приложений, использующих технологии машинного обучения, классификация изображений выделяется особой важностью. Она формирует основу для других задач компьютерного зрения и состоит в категоризации и присвоении меток группам пикселей или векторов в изображении по определенным правилам. Современные методы классификации изображений часто основаны на СНС (Convolutional Neural Networks, CNN), которые демонстрируют выдающуюся эффективность в распознавании и анализе визуальных данных.

Архитектура СНС представляет собой комплексную структуру слоев, спроектированных для обнаружения иерархических признаков в изображениях. Эффективная архитектура нейронной сети является ключевым фактором в достижении высокой точности классификации. С целью улучшения результатов, исследователи постоянно работают над созданием новых и оптимизацией существующих архитектур.

Актуальность темы исследования заключается в том, что классификация изображений находит все больше применений в связи с развитием информационных технологий, а СНС, благодаря своим характеристикам, являются одним из наиболее эффективных и поэтому популярных инструментов классификации изображений.

Основой для выполнения исследований являются методы, применяемые в теории распознавания образов, теории искусственного интеллекта, обработке больших данных, нейросетевом моделировании и разработке программного обеспечения.

Практическая значимость результатов исследования заключается в том, что они могут быть использованы для изучения имеющихся и разработки новых методов решения рассматриваемых задач.

## **ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ**

### **Цель и задачи исследования**

*Цель* диссертационной работы – разработать алгоритм и программное средство для решения задачи анализа графических объектов.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) Осуществить аналитический обзор и постановку задачи.
- 2) Определить алгоритмы, методы и модели, которые послужат основой проекта.
- 3) Разработать проект программного обеспечения.
- 4) Разработать алгоритмы и произвести их программную реализацию.
- 5) Протестировать полученный программный продукт.

*Объектом* исследования являются сверточные нейронные сети (СНС), применяемые в задачах анализа графических объектов.

*Предметом* исследования является эффективность СНС в задаче классификации изображений.

Основной *гипотезой*, положенной в основу диссертационной работы, является возможность использования СНС для анализа графических объектов. СНС разработаны специально для работы с данными, имеющими пространственную или растровую структуру, такими как изображения. Такие сети подходят для улавливания локального контекста, когда поток информации в пространстве не прерывается, то есть ее носители находятся в непосредственной близости друг от друга.

### **Связь работы с приоритетными направлениями научных исследований**

Научная работа выполнялась в соответствии с приоритетным направлением научных исследований Республики Беларусь «технологии больших данных» (Указ Президента Республики Беларусь от 7 мая 2020 г. № 156 «О приоритетных направлениях научной, научно-технической и инновационной деятельности на 2021–2025 годы»).

### **Личный вклад соискателя**

Результаты, приведенные в диссертации, получены соискателем лично.

## **Апробация результатов диссертации**

Основные положения диссертации опубликованы в 3 печатных работах в сборниках статей международных научно-практических конференций МОЛОДЫЕ ИССЛЕДОВАТЕЛИ — СОВРЕМЕННОЙ РОССИИ (Петрозаводск, Россия, 2022); SUSTAINABLE DEVELOPMENT FORUM - 2023 (Петрозаводск, Россия, 2023); Научный потенциал - 2024 (Петрозаводск, Россия, 2024).

## **Структура и объем диссертации**

Диссертация состоит из введения, общей характеристики работы, пяти глав, заключения, списка использованных источников, списка публикаций автора и приложений. В первой главе представлен анализ предметной области, выявлены основные существующие проблемы в рамках тематики исследования, показаны направления их решения, произведен обзор литературы и выполнена постановка задачи. Вторая глава посвящена рассмотрению основных алгоритмов, методов и моделей, которые служат основой проекта. В третьей главе предложен проект программного обеспечения, определяющий функциональность программного средства, а также определены среда и средства разработки программного обеспечения и требования к программному средству. В четвертой главе описывается процесс разработки и программной реализации алгоритмов и модели СНС. В пятой главе представлены результаты тестирования основного функционала разработанного программного средства.

Общий объем работы составляет 64 страницы, из которых основного текста – 49 страниц, 30 рисунков на 24 страницах, список использованных источников из 34 наименований на 3 страницах и 2 приложения на 8 страницах.

## ОСНОВНОЕ СОДЕРЖАНИЕ

Во **введении** определена предметная область, указаны основные направления исследования, показана актуальность темы диссертационной работы, дана краткая характеристика исследуемых вопросов, обозначены степень разработанности темы, методы, являющиеся основой для выполнения исследований, и практическая значимость работы.

Вопросам распознавания графических образов посвящено множество исследований. Существует несколько основных методов распознавания и большое число их модификаций. Точные алгоритмические методы, основанные на анализе контуров и поиске ключевых характеристик изображений, были исследованы в работах таких авторов, как Ч. Бэббидж, Дж. Вейценбаум, Н. Винер, В. Глушков, Дж. Маккарти, М. Мински, А. Ньюэлл, С. Пейперт, Д. Перл, Г. Поспелов, Д. Поспелов, Ф. Розенблатт, Г. Саймон, П. Уинстон, В. Финн, Н. Хомский, К. Шеннон. Совместно с алгоритмическими методами развивались методы, основанные на использовании нейронных сетей, основным отличием которых является процесс обучения классифицирующего звена системы распознавания. При этом нет необходимости в описании точного алгоритма определения принадлежности распознаваемого объекта к заданному классу. Методы, основанные на нейросетевых моделях, исследованы в научных работах таких авторов, как Я. Лекун, Й. Бенджио, Дж. Хинтон, А. Крижевский, И. Сатскевер, М. Цилер, Р. Фергюс, Э. Циссерман, К. Симонян, К. Сегеди, К. Хи, А. Карпати.

В **первой главе** представлен анализ предметной области, выявлены основные существующие проблемы в рамках тематики исследования, показаны направления их решения, произведен обзор литературы и выполнена постановка задачи.

Классификация изображений полезна для решения задач, когда необходимо определить принадлежность объекта одной из нескольких групп. Существует множество различных алгоритмов классификации изображений, таких как машины опорных векторов (support vector machines, SVM), K-ближайших соседей (K-nearest neighbours, KNNs), искусственные нейронные сети. Среди глубоких нейронных сетей СНС продемонстрировали отличные результаты в задачах компьютерного зрения, особенно в классификации изображений.

Для обучения различных систем обработки изображений и моделей машинного обучения часто используется набор данных MNIST, который содержит 70000 полутонных изображений рукописных цифр размером  $28 \times 28$  пикселей и представляет собой хорошо структурированный набор данных для задач классификации изображений.

Информация о существующих аналогах разрабатываемой модели СНС представлена на веб-ресурсе Papers with Code (PWC) [7]. Данные аналоги так же решают задачу классификации изображений рукописных цифр набора MNIST. Эти модели отличаются архитектурой и параметрами, что отражается на их эффективности.

Исходя из особенностей рассмотренных аналогов, при разработке алгоритма и программного средства была поставлена следующая задача: разработать высокоточную и конкурентоспособную модель СНС для классификации изображений, а также программное средство, которое позволит продемонстрировать работу этой модели.

**Вторая глава** посвящена рассмотрению фундаментальных подходов к решению поставленной задачи, а также методов устранения возникающих при этом проблем.

СНС доказали свою высокую эффективность для таких задач, как классификация и сегментация изображений, и даже использовались в передовых приложениях, таких как беспилотные автомобили и медицинская визуализация.

Чтобы преодолеть проблему переобучения, используется слой исключения (dropout), который удаляет несколько нейронов из нейронной сети в процессе обучения, что приводит к уменьшению размера модели. Исключение приводит к повышению производительности модели машинного обучения, поскольку предотвращает переобучение за счет упрощения сети [3-А].

При обучении нейронных сетей одним из наиболее важных параметров является коэффициент обучения. Он определяет, насколько сильно веса сети будут изменяться в процессе обучения. Значение коэффициента обучения должно быть тщательно настроено, чтобы обеспечить эффективное обучение сети.

Наиболее важную роль в искусственных нейронных сетях играют методы оптимизации, которые существенно влияют на процесс обучения. Конечная точность в процессе обучения зависит от согласования значений весов искусственных нейронов с функцией потерь, которую с каждой эпохой необходимо минимизировать. Если оптимизация проходит быстро и сходится к глобальному минимуму, то повышается точность распознавания и сокращается время обучения.

Для улучшения обобщающей способности нейронной сети, а также экономии памяти и ускорения вычислений используется преобразование входных изображений в оттенки серого.

Для управления переобучением модели используется регуляризация. Она помогает избежать слишком сложной модели, которая может хорошо подстроиться под обучающие данные, но будет работать плохо на новых данных.

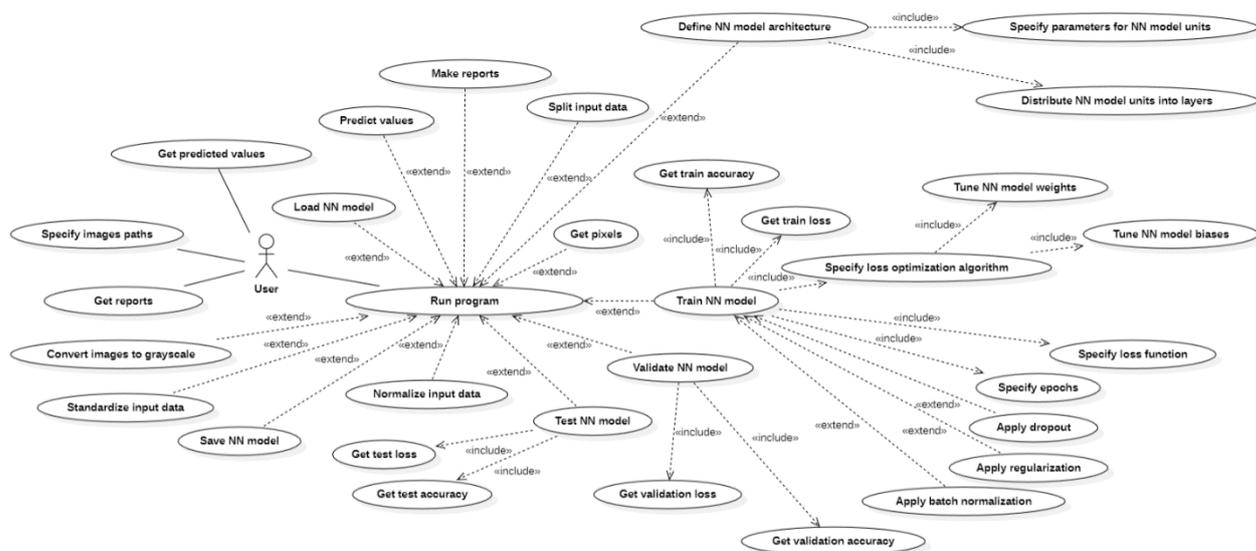
В задачах машинного обучения исходные данные часто заданы в разных единицах измерения и типах шкал. Такие данные следует преобразовывать в единое представление путем их нормализации или стандартизации. На сегодняшний день универсального метода нормализации данных, превосходящего другие методы, не существует, но нормализация исходных данных позволяет повысить точность их классификации.

Ключевое повышение производительности СНС произошло в основном за счет реорганизации блоков обработки данных, а также разработки новых блоков. Самые последние разработки в архитектуре СНС были достигнуты за счет использования глубины сети. Наиболее часто используемыми архитектурами СНС для классификации изображений являются LeNet, AlexNet, ZFNet, GoogLeNet, VGGNet, MobileNets и ResNet.

В третьей главе предложен проект программного обеспечения, определяющий функциональность программного средства, а также определены среда и средства разработки программного обеспечения и требования к программному средству.

Для описания возможного сценария использования разрабатываемой системы используется диаграмма вариантов использования (Use Case diagram) – одна из разновидностей диаграммы UML. Иными словами, она демонстрирует, как потенциальный пользователь будет взаимодействовать с системой. При том, что она отражает работу системы и то, какие ее функции будут актуальны для тех или иных пользователей, она не учитывает какие бы то ни было детали реализации (структуры данных и алгоритмы) [11].

Диаграмма вариантов использования проектируемого программного средства представлена на рисунке 1.



**Рисунок 1 – Диаграмма вариантов использования проектируемого программного средства**

Выбор инструментов разработки обусловлен спецификой предметной области.

Для взаимодействия пользователя с программным обеспечением используется графический пользовательский интерфейс.

Разработка программного обеспечения производилась в среде операционной системы Windows 10 и среде разработки PyCharm.

Для разработки программного средства использовался язык программирования Python и его модули. Python широко используется в разработке программного обеспечения для решения задач науки о данных и машинного обучения [15]. Python эффективен и работает на разных платформах. Программы на языке Python можно скачать бесплатно, они совместимы со всеми типами систем и повышают скорость разработки.

В **четвертой главе** описывается структура программного проекта, содержимое разработанных модулей и порядок их взаимодействия, общая организация работы программного средства, а также реализованная модель СНС.

Для обучения модели в качестве оптимизатора используется модификация алгоритма оптимизации Adam AMSGrad, предназначенная для улучшения свойств сходимости [19]. Основная идея AMSGrad заключается в использовании максимума прошлых квадратов градиентов для обновления параметров, вместо экспоненциального скользящего среднего [20], что помогает избежать резких изменений скорости обучения для каждой переменной.

В качестве функции потерь выступает категориальная перекрестная энтропия (categorical cross-entropy). Эта функция потерь работает с категориальным представлением данных, которое заключается в том, что метки классов описываются в виде векторов, длина которых равна числу классов [9]. Значения элементов вектора определяют принадлежность объекта к соответствующим классам.

Для более эффективного и стабильного обучения модели обучающий набор данных разбивается на пакеты по 256 примеров. Такой подход имеет ряд преимуществ, среди которых ускорение процесса обучения, уменьшение переобучения, улучшение обобщающей способности модели.

Модель СНС имеет следующую архитектуру (рисунок 2):

- Сверточный слой с 32 фильтрами размером  $3 \times 3$ , функцией активации ReLU и добавлением дополнительных пикселей по краю изображения.

- Сверточный слой с 48 фильтрами размером  $3 \times 3$ , функцией активации ReLU и добавлением дополнительных пикселей по краю изображения.

- Слой объединения со взятием максимума в окне с размером фильтра по умолчанию ( $2 \times 2$ ), шагом 2 по вертикали и 2 по горизонтали и без добавления дополнительных пикселей по краю изображения (по умолчанию).

- Сверточный слой с 48 фильтрами размером  $3 \times 3$ , функцией активации ReLU и добавлением дополнительных пикселей по краю изображения.
- Сверточный слой с 64 фильтрами размером  $3 \times 3$ , функцией активации ReLU и добавлением дополнительных пикселей по краю изображения.
- Слой объединения со взятием максимума в окне с размером фильтра по умолчанию ( $2 \times 2$ ), шагом 2 по вертикали и 2 по горизонтали и без добавления дополнительных пикселей по краю изображения (по умолчанию).
- Слой у плоского, преобразующий многомерный тензор от предыдущего слоя в линейный вектор для следующего.
- Полносвязный слой с 256 нейронами и функцией активации ReLU.
- Слой исключения, который случайным образом удаляет 30% нейронов из нейронной сети в процессе обучения.
- Слой пакетной нормализации, применяющий пакетное Z-преобразование к данным от предыдущего слоя.
- Выходной слой с 10 нейронами и функцией активации Softmax.

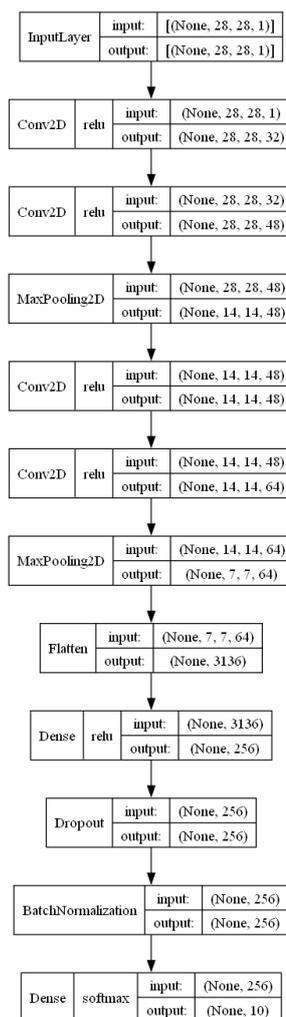
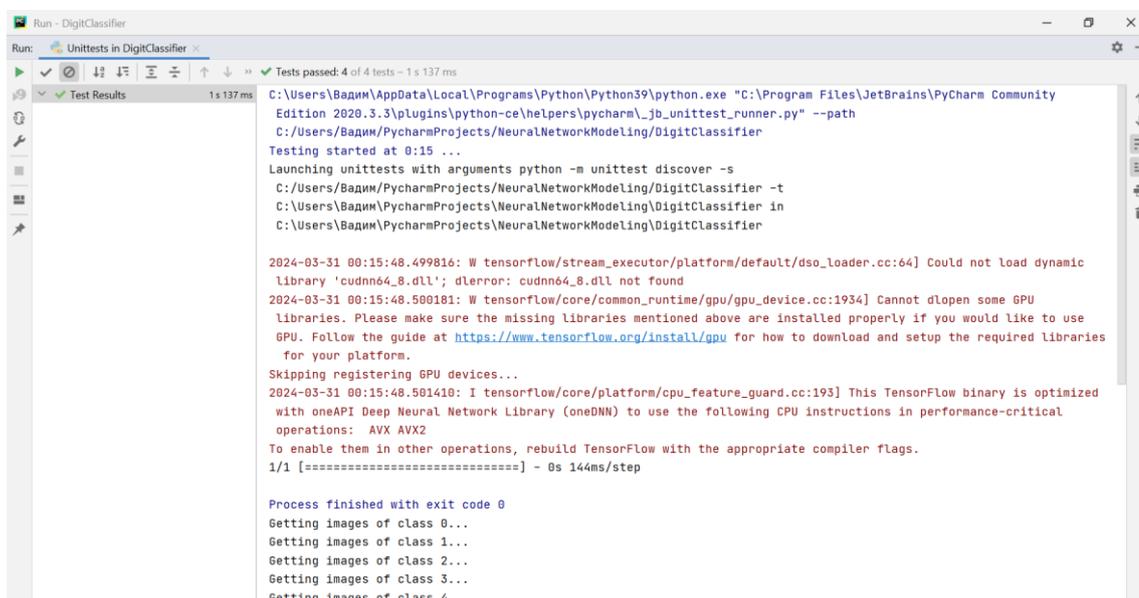


Рисунок 2 – Архитектура модели СНС

Обученная модель сохраняется в формате H5. Это расширение для бинарных файлов формата HDF5 (Hierarchical Data Format v.5), который используется для хранения и организации больших объемов данных [25, 26]. В частности, он позволяет эффективно хранить такие сложные данные, как архитектуру модели и ее веса. В библиотеках машинного обучения, таких как TensorFlow и Keras, существуют функции для сериализации и десериализации моделей в этом формате [27].

В пятой главе описываются методы и средства тестирования разработанного программного средства.

По результатам тестирования, представленным на рисунках 3–4, можно отметить корректную работу анализируемого основного функционала разработанного программного средства.



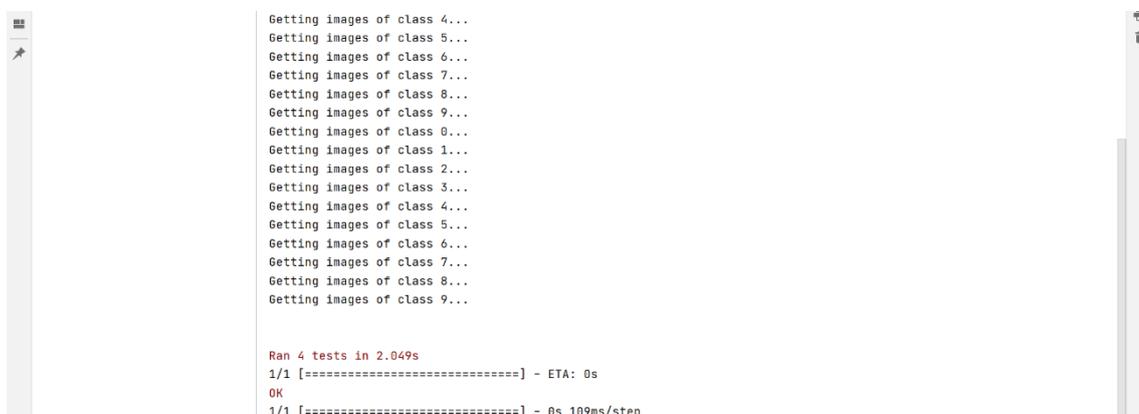
```
Run - DigitClassifier
Run: Unittests in DigitClassifier
Tests passed: 4 of 4 tests - 1 s 137 ms

Test Results 1 s 137 ms
C:\Users\Вадим\AppData\Local\Programs\Python\Python39\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2020.3.3\plugins\python-ce\helpers\pycharm\_jb_unittest_runner.py" --path
C:/Users/Вадим/PycharmProjects/NeuralNetworkModeling/DigitClassifier
Testing started at 0:15 ...
Launching unittests with arguments python -m unittest discover -s
C:/Users/Вадим/PycharmProjects/NeuralNetworkModeling/DigitClassifier -t
C:/Users/Вадим/PycharmProjects/NeuralNetworkModeling/DigitClassifier in
C:/Users/Вадим/PycharmProjects/NeuralNetworkModeling/DigitClassifier

2024-03-31 00:15:48.499816: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic
library 'cudnn64_8.dll'; dLError: cudnn64_8.dll not found
2024-03-31 00:15:48.500181: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1934] Cannot dlopen some GPU
libraries. Please make sure the missing libraries mentioned above are installed properly if you would like to use
GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries
for your platform.
Skipping registering GPU devices...
2024-03-31 00:15:48.501410: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized
with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical
operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
1/1 [=====] - 0s 144ms/step

Process finished with exit code 0
Getting images of class 0...
Getting images of class 1...
Getting images of class 2...
Getting images of class 3...
Getting images of class 4...
```

Рисунок 3 – Результаты тестирования



```
Getting images of class 4...
Getting images of class 5...
Getting images of class 6...
Getting images of class 7...
Getting images of class 8...
Getting images of class 9...
Getting images of class 0...
Getting images of class 1...
Getting images of class 2...
Getting images of class 3...
Getting images of class 4...
Getting images of class 5...
Getting images of class 6...
Getting images of class 7...
Getting images of class 8...
Getting images of class 9...

Ran 4 tests in 2.049s
1/1 [=====] - ETA: 0s
OK
1/1 [=====] - 0s 109ms/step
```

Рисунок 4 – Результаты тестирования

## ЗАКЛЮЧЕНИЕ

Анализ изображений – это извлечение значимой информации из изображений.

В настоящее время в области анализа графических объектов наибольший интерес представляют алгоритмы, основанные на методах машинного и глубокого обучения. С анализом изображений связана такая научная область, как компьютерное зрение.

Среди множества приложений, использующих технологии машинного обучения, классификация изображений выделяется особой важностью. Она формирует основу для других задач компьютерного зрения.

Классификация изображений полезна для решения задач, когда необходимо определить принадлежность объекта одной из нескольких групп. Существует множество различных алгоритмов классификации изображений, таких как машины опорных векторов (support vector machines, SVM), K-ближайших соседей (K-nearest neighbours, KNNs), искусственные нейронные сети. Среди глубоких нейронных сетей СНС продемонстрировали отличные результаты в задачах компьютерного зрения, особенно в классификации изображений.

СНС вычислительно эффективна. Ее можно использовать в различных отраслях для выполнения ключевых задач, таких как классификация изображений, распознавание лиц и анализ документов. Она выполняет совместное использование параметров и использует специальные алгоритмы свертки и объединения. На производительность СНС сильно влияет выбор гиперпараметров. Любое небольшое изменение значений гиперпараметров повлияет на общую производительность СНС. Поэтому тщательный выбор параметров является чрезвычайно важным вопросом, который следует учитывать при разработке схемы оптимизации.

Разработанная диаграмма вариантов использования позволила определить функциональность программного средства. Были определены среда и средства разработки программного обеспечения, а также требования к разрабатываемому программному средству.

Были разработаны и программно реализованы алгоритмы, отвечающие за взаимодействие пользователя с программным средством, и модель СНС, классифицирующая цифры. Также были реализованы алгоритмы, основанные на уже существующих в библиотеках Python и позволяющие осуществить этапы обучения, валидации и тестирования данной модели. По результатам выполнения этих этапов создаются и строятся соответствующие отчеты и графики, позволяющие оценить эффективность реализованной модели.

По результатам тестирования можно отметить корректную работу анализируемого основного функционала разработанного программного средства.

В результате исследования были разработаны высокоточная и конкурентоспособная модель СНС для классификации изображений, а также программное средство, которое позволяет продемонстрировать работу этой модели. Точность реализованной модели, оцененная на тестовом наборе из 45000 изображений рукописных цифр из набора данных MNIST, составляет 0,99489, что превосходит точность распознавания человеком рукописных цифр из этого набора данных, которая, согласно источнику [34], составляет 0,98. В силу двусмысленности интерпретации некоторых примеров из набора изображений трудно достичь точности 1. Согласно сведениям, представленным на ресурсе PWC, разработанная модель попадает по значению функции потерь в десятку лучших моделей из соответствующего рейтинга, а по точности – в двадцатку лучших моделей.

Использование современных технологий позволило сократить время на разработку программного средства, а также сформировать легкую для понимания структуру проекта, что облегчает возможность дальнейшей поддержки данного программного средства и предоставляет перспективы его усовершенствования путем расширения функционала.

Диссертационная работа является завершенной. Цель диссертационной работы была достигнута выполнением всех поставленных задач. Проверяемая гипотеза подтверждается полученными результатами. Результаты исследования могут быть использованы для изучения имеющихся и разработки новых методов решения рассматриваемых задач.

## СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

1-А. Венгеренко, В. В. Алгоритмы анализа графических объектов / В. В. Венгеренко, Н. Л. Боброва // МОЛОДЫЕ ИССЛЕДОВАТЕЛИ — СОВРЕМЕННОЙ РОССИИ : сб. ст. Междунар. науч.-практ. конф. (29 ноября 2022 г.). – Петрозаводск : МЦНП «Новая наука», 2022. – С. 121–127.

2-А. Венгеренко, В. В. Применение сверточных нейронных сетей для классификации изображений / В. В. Венгеренко, Н. Л. Боброва // SUSTAINABLE DEVELOPMENT FORUM - 2023 : сб. ст. II Междунар. науч.-практ. конф. (4 мая 2023 г.). – Петрозаводск : МЦНП «НОВАЯ НАУКА», 2023. – С. 100–109.

3-А. Венгеренко, В. В. Повышение эффективности обучения нейронных сетей / В. В. Венгеренко, Н. Л. Боброва // Научный потенциал - 2024 : сб. ст. Междунар. науч.-практ. конф. (14 марта 2024 г.). – Петрозаводск : МЦНП «НОВАЯ НАУКА», 2024. – С. 65–73.