**Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И.Ульянова (Ленина)**

**(СПбГЭТУ «ЛЭТИ»)**

|  |  |  |
| --- | --- | --- |
| **Направление** | 27.03.04 - Управление в технических системах | |
| **Профиль** | Без профиля | |
| **Факультет** | КТИ | |
| **Кафедра** | АПУ | |
| *К защите допустить* |  | |
| Заведующий кафедрой |  | Шестопалов М.Ю. |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**БАКАЛАВРА**

**Тема: Разработка алгоритмов анализа биомедицинских изображений**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент |  |  |  | Логинов Е.П. |
|  |  | *подпись* |  |  |
| Руководитель | доцент, к.т.н. |  |  | Каплун Д.И. |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |
| Консультанты | доцент, к.т.н. |  |  | Белаш О.Ю. |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |
|  | старший преподаватель |  |  | Лебедева Т.Н. |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |

Санкт-Петербург

2022

**ЗАДАНИЕ**

**на выпускную квалификационную работу**

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой АПУ |
|  | \_\_\_\_\_\_\_\_\_\_\_\_ Шестопалов М.Ю. |
|  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Студент | Логинов Е.П. | | | |  | Группа | 7896 |
| Тема работы: Разработка алгоритмов анализа биомедицинских изображений | | | | | | | |
| Место выполнения: СПбГЭТУ «ЛЭТИ» Факультет КТИ Кафедра АПУ | | | | | | | |
| Исходные данные (технические требования):  Разработка ансамбля нейронных сетей с целью повышения качества классификации биомедицинских изображений. | | | | | | | |
| Содержание ВКР:  Аналитический обзор (описание архитектур нейронных сетей, используемых для решения задач классификации изображений, принципов установки параметров нейронных сетей), описание модели, описание модели ансамбля нейронных сетей, экономическое обоснование. | | | | | | | |
| Перечень отчетных материалов: текст ВКР, иллюстративный материал | | | | | | | |
| Дополнительные разделы: Экономическое обоснование | | | | | | | |
| Дата выдачи задания | | | Дата представления ВКР к защите | | | | |
| «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. | | | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. | | | | |
| Студент | |  | | Логинов Е.П. | | | |
| Руководитель доцент, к.т.н. | |  | | Каплун Д.И. | | | |
| *(Уч. степень, уч. звание)* | |  | |  | | | |

**календарный план выполнения**

**выпускной квалификационной работы**

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой АПУ |
|  | \_\_\_\_\_\_\_\_\_\_\_\_ Шестопалов М.Ю. |
|  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | Логинов Е.П. |  | Группа | 7896 |
| Тема работы: Разработка алгоритмов анализа биомедицинских изображений | | | | |

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование работ | Срок выполнения |
| 1 | Обзор литературы по теме работы | 08.02 – 21.02 |
| 2 | Формирование набора медицинских изображений | 22.02 – 26.02 |
| 3 | Анализ архитектур и показателей качества нейронных сетей | 01.03 – 28.03 |
| 4 | Разработка и настройка ансамбля нейронных сетей | 05.04 – 25.04 |
| 5 | Оформление пояснительной записки | 03.05 – 23.05 |
| 6 | Оформление иллюстративного материала | 24.05 – 30.05 |

|  |  |  |
| --- | --- | --- |
| Студент |  | Логинов Е.П. |
| Руководитель доцент, к.т.н. |  | Каплун Д.И. |
| *(Уч. степень, уч. звание)* |  |  |

**РЕФЕРАТ**

**Пояснительная записка 81 стр., 76 рис., 5 табл., 21 ист.**

СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ, АнСАМБЛЬ НЕЙРОННЫХ СЕТЕЙ, РЕНТГЕНОГРАММА ГРУДНОГО ОТДЕЛА, КЛАССИФИКАЦИЯ ИЗОБРАЖЕНИЙ, КОРОНОВИРУСНАЯ ИНФЕКЦИЯ

Объектом исследования является ансамбль нейронных сетей для классификации рентгеновских снимков грудного отдела.

Цель работы – изучение существующих архитектур нейронных сетей, методов повышения качественных показателей нейронных сетей в задачах классификации изображений. Моделирование ансамбля нейронных сетей для повышения точности определения класса изображения.

В настоящей работе описаны существующие методы решения задачи классификации изображения. Дано краткое описание основных параметров нейронных сетей (оптимизатор, функция потерь, функция активации) и архитектуры нейронных сетей (базовая модель, сверточный слой, полносвязный слой, выходной слой).

В результате разработана модель ансамбля нейронных сетей с большим значением точности классификации биомедицинских изображений.

**ABSTRACT**

In this paper, described a Deep Convolutional Neural Network-based solution which can detect the COVID-19 positive patients using chest X-Ray images. Multiple state-of-the-art CNN models — Resnet50V2, Xception and Inceptionv3, have been adopted in the proposed work. They have been trained individually to make independent predictions. Then the models are combined, using a new method of **weighted average ensembling** technique, to predict a class value. To test the efficacy of the solution publicly available chest X-ray images of [COVID-19 Radiography Dataset](https://www.kaggle.com/code/gpiosenka/feature-rich-callback-f1-score-95/data) was used. To make dataset balanced extra dataset of Chest X-Ray Images (Pneumonia) was concatenated to initial dataset. Total dataset have been divided into training and test sets. The proposed approach gave a classification accuracy **of 91.62%** which is higher than the state-of-the-art CNN models as well the compared benchmark algorithm.

**СОДЕРЖАНИЕ**

[Введение 7](#_Toc105888491)

[1 Аналитический обзор 8](#_Toc105888492)

[1.1 Развитие нейронных сетей 8](#_Toc105888493)

[1.2 Сверточная нейронная сеть 8](#_Toc105888494)

[1.3 Модели СНН для классификации рентгенограмм 9](#_Toc105888495)

[1.4 Проблематика данных 16](#_Toc105888496)

[1.5 Безопасность СНН 18](#_Toc105888497)

[1.6 Архитектура СНН 19](#_Toc105888498)

[1.7 Параметры нейронных сетей 27](#_Toc105888499)

[1.8 Методы построения ансамбля 35](#_Toc105888500)

[2 Второй раздел 40](#_Toc105888501)

[3 Третий раздел 41](#_Toc105888502)

[4 Экономическое обоснование 42](#_Toc105888503)

[Заключение 49](#_Toc105888504)

[Список используемых источников 50](#_Toc105888505)

# Введение

Разработка алгоритмов анализа биомедицинских изображений в настоящее время ведется многими исследователями.

По мере развития математического аппарата, совершенствования полупроводниковых технологий и роста объемов доступных данных, реализуются новые подходы к обучению нейронных сетей с целью ускорения обучения и одновременного повышения качества и точности результатов, получаемых по итогу валидации модели. На текущий момент существует несколько моделей нейронных сетей, которые показывают более 90% точности в задачах классификации изображений. Определение архитектуры используемой нейронной сети, её параметров, в основном зависит от опыта разработчика модели и характеристик набора изображений, на которых будет обучаться и тестироваться нейронная сеть.

В работе приведено описание моделей нейронных сетей по обработке рентгенограмм грудной клетки в порядке их хронологического появления, с описанием применяемых методов обучения и подходов, призванных к повышению качественных характеристик по классификации рентгеновских снимков[1-3].

Раскрыта проблематика накопления, обработки и передачи данных для обучения и тестирования разрабатываемых моделей. Представлены существующие подходы к решению проблемы качества датасетов, как с помощью технической обработки имеющихся наборов данных, так и организационного характера – взаимодействия многих исследователей и пользователей данных.

Основная цель работы заключается в объединении трех моделей сверточных нейронных сетей в ансамбль и выработки правила получения более точного результата. Ожидается, что предложенный метод повысит точность классификации изображений.

1. Аналитический обзор

## Развитие нейронных сетей

Модели глубокого обучения состоят из глубоких нейронных сетей, которые автоматически извлекают важную информацию из данных. Обучение глубоких моделей обычно выполняется с использованием оптимизации стохастического градиентного спуска.

За последнее десятилетие машинное обучение и искусственный интеллект получили значительное развитие, как техническое, так и математическое. Нейронные сети превзошли традиционные модели по многим важным показателям. Зарекомендовали себя в широком спектре областей – распознавание речи, обработка изображений, распознавание речи.

Качество анализа и классификации медицинских изображений уже можно де факто признать стандартом. Сверточные нейронные сети (СНН) использовались для различных задач классификации медицинских изображений, таких как: болезни легких, определение переносчика малярии в образцах крови, рак груди, беспроводная эндоскопия, диагностик рака кожи по классификации изображения.

Исследование по валидации и адаптируемости методов глубокого анализа для выявления COVID-19 с использованием классификации рентгеновских изображений грудной клетки показало результаты с точностью 95,12%, чувствительностью 97,91% и специфичностью 91,87%.[1]

На невысокие качественные характеристики ранних моделей влияли недостаточность снимков с COVID-19 и низкое качество изображений в некоторых датасетах.

## Сверточная нейронная сеть

Нейронная сеть имитирует работу части человеческого мозга, отвечающую за распознавание образов. Модель нейронная сети используется для распознавания и анализа изображений, классификации изображений, а так же распознавания речи[4]. СНН это тип глубоких нейронных сетей, содержащий сверточные, объединяющие уровни, и уровень активации. Основным является уровень свертки, в котором расположены ядра, на входы которым подается изображение. Все выходы сверточного слоя составляют карту объектов.[1]

Двухразмерные СНН применяются в задачах классификации изображений, а также определения объектов на изображении, сегментации изображений, распознавания лица.

Классификация изображений - это задача отнесения изображения к классовой категории.

Нейронная сеть структуры LeNet-5 считается первым приложением, используемым для классификации рукописных цифр. AlexNet заставил подходы к классификации, основанные на СНН, дал толчок к дальнейшему развитию направления. Исследователи отметили важность глубины нейронной сети, но эти первые СНН состоят не более чем из десяти слоев. Впоследствии появились более глубокие сетевые структуры, такие как GoogLeNet и VGGNets, которые значительно повышают точность в задачах классификации[4].

## Модели СНН для классификации рентгенограмм

**COVID‑Net**

Одна из первых моделей сверточных нейронных сетей c открытым исходным кодом, настроенная для выявления COVID-19 на рентгеновских снимках грудной области[3]. Реализована в 2020 году.

Открыла новое применение искусственного интеллекта для раннего, эффективного и масштабного выявления вируса среди людей.

Использует датасет COVIDx, содержащий 13 975 изображений от 13 870 пациентов. Датасет является компиляцией пяти открытых источников данных:

1. COVID-19 Image Data Collection
2. COVID-19 Chest X-ray Dataset Initiative
3. ActualMed COVID-19 Chest X-ray Dataset Initiative
4. RSNA Pneumonia Detection Challenge
5. COVID-19 radiography.

Модель предобучена на наборе данных ImageNet, а затем обучена на COVIDx датасете с использованием Adam оптимизатора. Коэффициент скорости обучения установлен , число эпох обучения 22, размер батча 64. Первоначальный прототип COVID-Net был создан и оценен с использованием библиотеки глубокого обучения Keras с серверной частью Tensor-Flow.

**Сверточная нейронная сеть для определения COVID-19**

Модель тестировалась на 100 снимках для получения точности 100%. Затем модель была протестирована на несвязанном датасете и показала 99,5% точности[2]. Реализована в 2021 году.

Для обучения модели использовался датасет *“Github COVID19 X-ray dataset”* из репозитрия *https://github.com/ieee8023/COVID-chestxray-dataset*. Поскольку размер датасета мал и несбалансирован, то для решения этой проблемы была применена аугментация. Для генерации широкого набора вариантов применялись поворот изображения на разные углы. Для балансировки датасета по классам были добавлены изображения минорных классов.

Первый этап - сбор первичного набора изображений двух классов: с COVID-19 и здоровых пациентов. Вторым этапом некачественные и недостаточно информативные для постановки диагноза изображения исключались медицинскими специалистами. Таким образом был сформирован качественный датасет. На третьем этапе датасет был аугментирован стандартными техниками для увеличения его размера. Результирующий датасет был использован для обучения сети на последующем этапе. После обучения модель была протестирована на качество распознавания на первичном наборе данных, а также на обособленном датасете. Каждый датасет имел свои характеристики по количеству изображений, тестовый набор, сверочный набор, и разное соотношение изображений, принадлежащих одному классу.

В первичный набор состоял из 178 рентгеновских изображений грудной клетки. Из которых 136 были снимками с подтвержденным заболеванием, и 42 изображения - обычные или с различными заболеваниями, такими как пневмония (*J. P. Cohen, “Github COVID19 X-ray dataset,” 2020, https://github.com/ieee8023/COVID-chestxray-dataset, 2020*.). Таким образом, датасет состоял из изображений двух классов – COVID-19 и другие. Такое распределение показывает несбалансированность датасета. Следовательно, для достижения лучшего результата необходима предподготовка. На изначальном наборе данных модель показала недостаточную точность в 54%.

Для балансировки набора данных использовались изображения 136 изображений без признаков заболевания из набора данных (*P. Mooney, “Kaggle X rays dataset,” 2020, https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia Online*.)

После балансировки датасета модель показала точность в 69%, что еще не достаточно эффективной системы диагностики COVID-19.

Тщательный анализ изображений медицинскими специалистами 135 изображений выделил 90 изображений, подходящих для обучения модели. Результирующий датасет содержал 90 изображений COVID-19 и 90 обычных изображений. Полученный результат показал улучшение качества распознавания заболевания в 72%. Однако, невысокий прирост качественной характеристики обусловлен небольшим количеством изображений.

Аугментация данных, это способ существенно увеличить объем данных для обучения модели. Для изображений используются базовые операции такие как переворачивание, вращение, обрезка или дополнение. Первоначальный набор данных дополняется измененными таким образом изображениями, существенно увеличивая размер датасета для обучения нейронной сети. Изменение изображений дает дополнительные данные для обучения модели.

Модель состоит из 38 слоев, в каждом из которых 6 сверточных, 6 объединяющих, 6 дропаута, 8 слоев с функциями активации, 8 слоев батч нормализации, 1 сглаживающий и 3 полносвязных слоя. Размер входных изображений 150\*150\*3 – 150\*150 в цвете RGB.

Подход к определению числа слоев состоял в постоянном увеличении количества слоев, до достижения необходимых результатов, начиная с одного слоя.

**AC-COVIDNet**

Модель изучает устойчивые и дискриминационные признаки с помощью контрастных потерь. Более того, предлагаемый метод придает большее значение инфицированным областям, управляемым механизмом внимания[3]. Описана в 2022 году.

Особенности:

* Используется модуль внимания, помогающий обучению.
* Контрастные потери увеличивают способность к классификации изображения и надежность модели.

Использовались возможности механизма внимания и контрастного обучения, чтобы справиться с обучением с ограниченными данными для распознавания COVID-19 по изображениям.

Механизм внимания в глубоком обучении облегчает изучение локализованных признаков, что более важно в контексте проблемы распознавания COVID-19 на рентгенограммах.

Также обнаружено, что модель, основанная на внимании, может превзойти модели простой нейронной сети.

Сравнительное обучение - это свежая тенденция к изучению сходства и различия между похожими и непохожими образцами в абстрактном пространстве признаков для визуальных представлений. Как правило, контрастивное обучение зависит от сходства признаков между положительными и отрицательными парами.

Модель основана на модели COVID-Net. В модели широко используется легкий механизм расширения проекции остаточного проецирования (PEPX). В этой архитектуре также используется селективная связь на большие расстояния в модели, что улучшает репрезентативность. Это также облегчает обучение модели. Однако широкое использование соединений дальнего действия может привести к появлению множества избыточных низкоуровневых признаков. Для решения этой проблемы в предлагаемой модели используется механизм внимания. Внимание помогает модели расставить приоритеты в важных областях. Внимание также полезно для подавления активации избыточных функций из начальных слоев и помогает сосредоточиться на важных функциях, необходимых для решения задачи. Шлюзы внимания используются в предлагаемой архитектуре на различных уровнях архитектуры COVID-Net, где используется много соединений дальнего действия. Это повышает чувствительность, поскольку модель лучше учитывает важные визуальные особенности зараженных областей на изображениях. Поскольку разница между признаками COVID-19 и пневмонии очень тонкая, предлагается использовать контролируемую контрастную потерю. Контрастная потеря помогает сети максимально увеличить расстояние между выученными представлениями классов.

Архитектура проекции-расширения-проекции-расширения (PEPX)[3]. Задача модуля состоит в том, чтобы проецировать объекты в более низкое измерение, используя первые два слоя *conv 1x1*, затем расширять эти объекты с помощью слоя свертки по глубине (*DWConv3x3*) и снова проецировать в более низкое измерение, используя два слоя *conv 1x1*. Таким образом, уровень PEPX приводит к эффективной модели за счет уменьшения количества параметров и операций.

Ворота внимания используются на различных уровнях. Выделенные объекты из нескольких слоев проходят через *conv1x1* и складываются вместе. Затем агрегированные функции передаются через функцию активации *Relu*, за которой следует *conv 1x1*, а затем функция активации сигмовидной формы. Затем выходные данные карты объектов сигмовидного слоя проходят через ресемплер. Выходные данные ресемплера добавляются к функциям из ближайшего входного слоя к модулю внимания, чтобы получить выход ворот внимания (рисунок 1).



Рисунок 1 - Схема модуля Ворота внимания[3]

Модель обучалась на данных датасета COVIDx.Это самый большой публичный набор рентгенограмм специализированный под COVID-19. Этот датасет является компиляцией нескольких открытых наборов изображений. Содержит 14 258 изображений трех классов: «COVID-19», «пневмония», и «без патологий», распределенных следующим образом:

* 7 966-без патологий
* 5 475 –пневмония
* 517 – COVID-19

в обучающем наборе и по 100 изображений каждого класса для тестирования.

Последующее обучение модели состоит из двух этапов. На первом этапе сеть обучается с использованием функции потерь, для извлечения объектов. На втором этапе экстрактор признаков фиксируется и обучается путем добавления классификатора с функцией перекрестной потери энтропии.

Модель обучалась на трех созданных вариантах набора изображений. Использовался оптимизатор Adam. Значение уровня обучения составляло . Размер батча – 64. На каждом слое использовалась функция активации *ReLU* и софтмакс на последнем слое. Пуллинг с параметром max используется после каждого PEPX слоя.

**Использование ансамбля нейронных сетей**

В основном исследователи используют одну архитектуру нейронной сети и с помощью подбора количества слоев и настройки коэффициентов пытаются достичь необходимых показателей качества распознавания (классификации) изображения. Некоторые исследователи создают новые архитектуры для решения задачи. Ни одно из ранних исследований не пыталось комбинировать модели с целью улучшения качества классификации модели в целом[5]. Несколько исследований показывают, что ансамблевые модели лучше предсказывают, чем индивидуальные модели, а также помогают предотвратить переобучение. Одним из методов ансамблирования является вычисление средневзвешенного значения выходной вероятности каждой нейронной сети. Такой метод является более качественным, по сравнению с методами, основанными на простом среднем значении.

Для принятия взвешенного решения в реальной жизни во многих сферах основываются на несколько мнений. Используя этот подход, были адаптированы несколько моделей нейронных сетей. Каждая модель обучалась независимо. После обучения модели были скомбинированы, с использованием средневзвешенной средней для предсказания класса.

В модели используются три сверточные нейронные сети - DenseNet201, Resnet50V2, Inceptionv3.

Особенностью DenseNet является меньшее количество настраиваемых параметров, чем в других сверточных нейронных сетях. Каждый слой сети принимает на вход карты объектов от всех предыдущих слоев. Такая передача позволяет качественней передавать объекты между слоями.

Сеть ResNet50V2 с помощью остаточных блоков решает проблему исчезновения градиента.

Inception мощная модель, с высокой точностью извлечения признаков и классификации изображения на основе этих признаков. Модель состоит из 48 слоев, в каждом из которых 11 модулей. Каждый модуль состоит из сверточных фильтров, пулинг слоев, и модуля функции активации ReLU. Значение дропаута – 0.6.

При распределении весов в ансамбле больший вес присваивается модели с меньшей ошибкой валидации.

Датасет для обучения ансамбля собран из других открытых наборов данных. Собранные наборы данных содержат три класса изображений – «COVID», «пневмония», «нормальный». Датасет разделен на два класса – «COVID+» и «COVID-». В итоге изображений класса «COVID+» в датасете 538 и 468 класса «COVID-».

Оценка эффективности модели основывается по следующим параметрам: точность классификации, чувствительность и цена *F1*.

Модель принятия решения, основанная на технике средневзвешенного решения, показала точность 91,62%, и чувствительность 95%.

**Выводы**

Развитие разработки СНН ведется по направлениям создания новых архитектур под конкретные задачи, настройки параметров контроля прогресса обучения, или же путем комбинирования и настройки нескольких моделей, и выработки результата на основе взвешенного подхода.

## Проблематика данных

**Качество данных**

Одним из этапов обучения нейронной сети является предобработка имеющихся данных. Необходимость предобработки связана с недостатком качественных, размеченных и подтвержденных данных.

При разработке каждой модели нейронной сети требуется размеченный тренировочный набор данных.

Общей проблемой данных является их доступный объем. Имеющиеся открытые датасеты имеют разные характеристики по объему, по качеству и размеру изображений, сбалансированность по классам.

Также существенным барьером для развития искусственного интеллекта является отсутствие качественной верификации данных. Процесс разработки качественной разметки тренировочного набора данных требует вовлечения экспертов в своей области. Разметка набора данных трудоемкий процесс, поэтому многие публично доступные наборы данных небольшого объема.

Каждый медицинский центр может создавать датасеты различного качества разметки и формата, что затрудняет процесс обучения.

Следовательно, качество обучения алгоритмов анализа медицинских изображений связано с качеством тренировочного датасета[6].

Таким образом, существует проблема сбора, верификации, накопления и предоставления данных.

В России для решения проблемы качественных данных в 2020 году приступили к формированию федеральных (общероссийских) датасетов экспертами Центра диагностики и телемедицины[7]. Первый датасет ориентирован на рентгенограммы и снимки компьютерной томографии пациентов с признаками COVID-19. Целью проекта является создание эталонного датасета, с помощью которого специалисты смогут оценивать корректность и точность работы сервисов по выявлению патологий в исследуемых областях.

Важной особенностью датасета является полная деперсонализация данных, то есть полное исключение персональных сведений о пациенте. Такой подход снимает вопрос защиты персональных данных при передаче накопленного массива исследователям и разработчикам.

В период пандемии исследователи Центра диагностики и телемедицины собрали самую большую в мире базу данных пациентов с признаками коронавирусной пневмонии. База собиралась в сжатые сроки, что было вызвано необходимостью автоматизации процесса распознавания патологических изменений в легких при коронавирусной инфекции. На основе датасета, представленного в открытом доступе, разработчики всего мира могли "дообучать" и тестировать алгоритмы своих сервисов.

**Выводы**

Создание больших по объему, качественных данных со стандартными характеристиками, предоставление свободного доступа к имеющимся наборам данных, позволит ускорить разработку и тестирование новых моделей. Качественный набор данных должен подкрепляться разметкой изображений экспертами. Применение единых стандартов к качеству и размеру изображений, позволит избежать трудоемкой разработки методик предварительной обработки изображений, а разрабатывать архитектуру нейронной сети.

## Безопасность СНН

Существует множество применений СНН в повседневной жизни, включая систему идентификации безопасности, идентификацию медицинских изображений, распознавание дорожных знаков и распознавание номерных знаков[4]. Эти приложения тесно связаны с безопасностью жизни и имущества. Как только модели будут нарушены или уничтожены, последствия будут серьезными. Поэтому безопасности моделей СНН следует придавать большое значение. Исследователи в сфере безопасности реализовали некоторые методы для обмана СНН, что привело к резкому падению точности. Эти методы можно разделить на две категории: «отравление» данными и враждебные атаки.

«Отравление» данными указывает на то, что происходит «отравление» обучающих данных на этапе обучения. «Отравление» относится к вставке данных о шуме в обучающие данные. Его нелегко отличить на уровне изображения, и в процессе обучения не обнаруживается никаких отклонений. Кроме того, данные о шуме могут быть даже точно настроены таким образом, чтобы модель могла неправильно идентифицировать определенные цели. Сгенерированные маски возмущений вводятся в обучающие выборки в качестве бэкдора для обмана модели. Внедрение бэкдора не мешает нормальному поведению, но стимулирует экземпляр бэкдора неправильно классифицировать конкретные цели[4].

Состязательная атака также является одной из угроз, с которыми сталкиваются глубокие нейронные сети. К обычному изображению добавляется некоторый шум. Хотя невооруженный глаз не может различить разницу между двумя изображениями, модель, основанная на СНН, не может распознать их как одно и то же. Исследователи считают, что основным фактором уязвимости нейронных сетей являются линейные характеристики, такие как ReLU, Maxout и другие. Разработчики перечислили три направления защиты от атак противника, которые соответственно улучшаются на обучающих примерах, модифицированных обученных сетях и дополнительных сетях. Во-первых, для обучающих примеров можно использовать состязательные примеры для повышения надежности моделей. Во-вторых, сетевую архитектуру можно настроить таким образом, чтобы она игнорировала шум. Наконец, дополнительные сети могут быть использованы для защиты магистральной сети от атак злоумышленника[4].

## Архитектура СНН

**Базовая модель сверточной нейронной сети**

Фактически, архитектура сверточной нейронной сети представляет собой нейронную сеть, содержащую несколько скрытых слоев, каждый из которых имеет несколько двумерных плоскостей, состоящих из нескольких нейронов[8]. Кроме того, все нейроны предполагаются независимыми. Его входные данные можно рассматривать как двумерное изображение, а модуль извлечения объектов встроен в архитектуру СНН. На рисунке 2 представлена базовая архитектура сверточной нейронной сети.

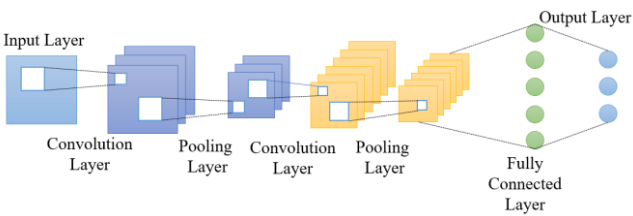


Рисунок 2 - Базовая архитектура сверточной нейронной сети[8]

Нейронную сеть обычно обучают на размеченных данных, то есть, пар векторов ввод-вывод

(1)

где используется в качестве входных данных, желаемый выходной сигнал нейронной сети. Цель обучения минимизировать значение функции потерь[9]

(2)

параметра , где - это функция потерь.

Наиболее популярным алгоритмом для поиска хороших наборов параметров является стохастический градиентный спуск (*SGD*), который начинается с некоторых случайных начальных значений , а затем итеративно обновляет

(3)

где - скорость обучения (*learning rate*) и - это аппроксимация функции потерь, которая вычисляется для случайной малой партии(*mini-batch*) обучающих примеров размера на каждой итерации.

Выбирая малым по сравнению с , сложность вычисления градиента значительно снижается при одновременном снижении дисперсии обновления веса. Cуществует множество вариантов алгоритма *SGD*, которые динамически адаптируют скорость обучения для улучшения сходимости.

**Входной слой**

Входной набор необработанных данных может быть непосредственно передан во входной слой. Одно изображение фактически вводится по его пиксельному значению во входной слой[8].

**Сверточный слой**

Также известен как уровень с расширенной выборкой, предназначен для извлечения объектов из входных данных. Свертка - это ключевой шаг для извлечения объектов. Каждый сверточный слой имеет свое собственное сверточное ядро, которое извлекает различные объекты из входных данных. Количество извлеченных объектов растет по мере увеличения числа сверточных ядер, включенных в слой[8]. Выходные данные свертки называются картами объектов.

Слои сверточной нейронной сети были описаны в работе *Y. LeCun et al., “Generalization and network design strategies,” Connectionism in perspective, pp. 143–155, 1989.*, чтобы обеспечить эффективный метод обучения для двумерных изображений. Связывая смежные сдвиги одних и тех же весов вместе способом, аналогичным тому, как фильтр скользит по входному вектору, сверточные слои способны изучать объекты с инвариантностью к сдвигам во входном векторе. Они также значительно снижают сложность модели, необходимую для представления эквивалентных объектов, инвариантных к сдвигу, с использованием полностью связанных слоев, уменьшая сложность оптимизации *SGD* и улучшая обобщение на соответствующих наборах данных[9].

В общем случае сверточный слой состоит из набора весов фильтров[9], которые генерируют для каждого так называемую карту объектов из входной матрицы.

Выходные размеры могут быть уменьшены либо путем увеличения шага сдвига фильтра, либо путем добавления объединяющего слоя (слой пуллинга).

При установке ядра свертки определенного размера может потеряться информация на краю изображения. Следовательно, вводится заполнение для увеличения входных данных с нулевым значением, которое может косвенно регулировать размер. Кроме того, для контроля плотности свертывания используется шаг. Чем больше шаг, тем ниже плотность. После свертки карты объектов состоят из большого количества объектов, что может привести к проблеме переобучения[4].

Существует множество различных методов свертки, таких как разделяемые свертки, групповые свертки и многомерные свертки[4].

**Слой пуллинга**

Его основная функция состоит в завершении второго извлечения данных объектов, и передачи следующему сверточному слою. При нормальных условиях архитектура СНН содержит, по меньшей мере, два сверточных слоя и два слоя с понижающей дискретизацией соответственно. Чем больше уровней архитектуры задано, тем больше вероятность того, что извлечение объектов из входных данных поможет решению задачи классификации[8].

Объединяющий слой разбивает карту объектов на области , для каждой из которых он вычисляет одно выходное значение, например, максимальное или среднее значение или L2-норму[9].

Применение этого слоя позволяет уменьшить размер карт объектов и параметров сети.

Этот слой служит двум основным целям: первый заключается в уменьшении количества параметров или весов, тем самым уменьшая вычислительные затраты, а второй - в контроле переобучения. Ожидается, что идеальный метод объединения будет извлекать только полезную информацию и отбрасывать ненужную[10].

**Методы пуллинга**

**Среднее объединение**

Идея среднего или среднего значения для объединения и извлечения признаков, впервые представленная в *LeCun, Y., et al. Handwritten digit recognition with a back-propagation network. in Advances in neural information processing systems. 1990. и используемая в LeCun, Y., et al., Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998*. Это первая глубокая нейронная сеть, основанная на свертке. Как показано на рисунке 3, средний уровень объединения выполняет выборку с понижением, разделяя входные данные на прямоугольные области объединения и вычисляя средние значения каждой области[10].

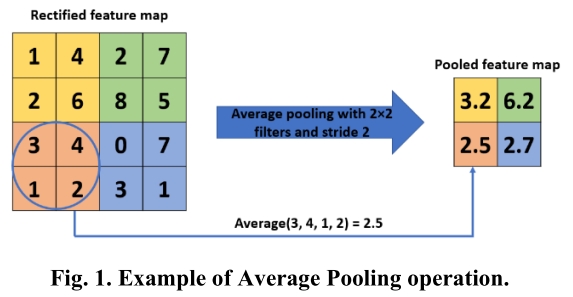


Рисунок 3 - Пуллинг по среднему значению[10]

**Максимальное объединение**

Оператор максимального объединения может быть применен для уменьшения выборки сверточных выходных полос, тем самым уменьшая изменчивость[10].

На рисунке 4 представлен пример свертки по максимальному значению.

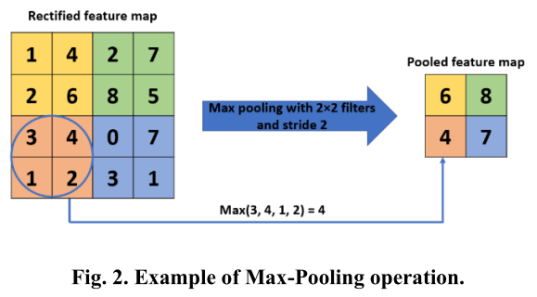


Рисунок 4 - Пуллинг по максимальному значению[10]

Также существует несколько других и новаторских методов реализации этого слоя, такие как смешанный пуллинг, *-pooling*, *Spatial Pyramid Pooling*.

**Полносвязный слой (*FC*)**

Все карты объектов соединяются вместе в качестве входных данных и представляют вектор размерности n\*1, где n это количество фильтров финального слоя. В общем случае узлы нейронов в более позднем слое соединены с узлами нейронов в предыдущем слое, но узлы в каждом слое разъединены. Этот слой интегрирует и нормализует абстрагированные функции предыдущих сверток, чтобы получить вероятность для различных условий.

В типичной глубокой нейронной сети слои FC содержат большинство параметров сети. На рисунке 5 представлен график числа параметров для архитектур нейронных сетей, показавших лучшие результаты в ILSVRC[11].

Несмотря на их распространенность, гиперпараметры, такие как количество слоев FC и количество нейронов, требуемых в слоях FC для данной архитектуры СНН для получения лучшей производительности, не изучены[12].

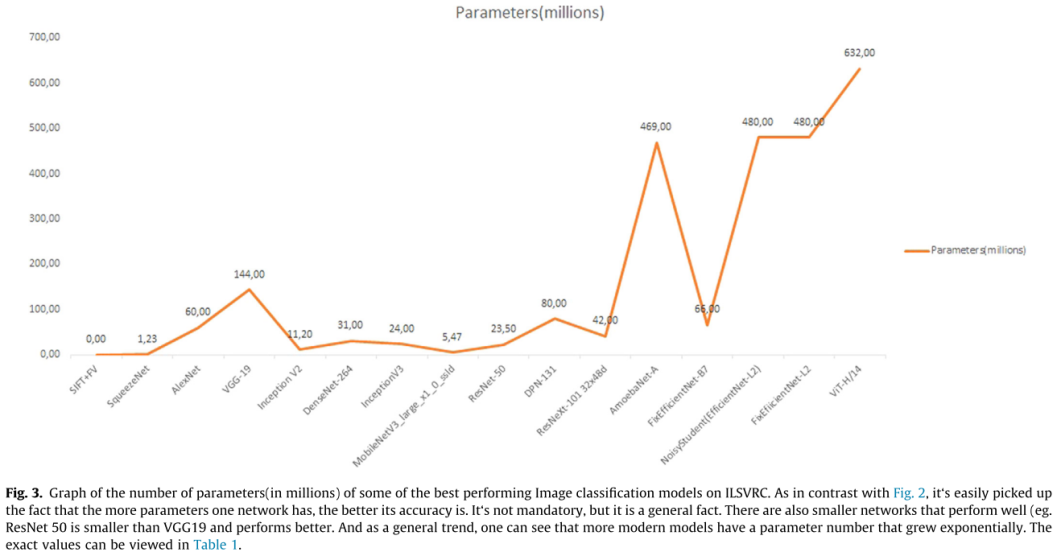


Рисунок - Число параметров в нейронных сетях различных архитектур

Это огромное количество обучаемых параметров в слоях FC требуется для подгонки сложных нелинейных функций в пространстве признаков, в которое отображаются вводимые элементы данных. Однако такое большое количество параметров может привести к чрезмерной подгонке классификатора[12].

Чтобы получить более высокую производительность, мелкие СНН требуют большего количества узлов в слоях FC. С другой стороны, более глубоким СНН требуется меньшее количество нейронов в слоях FC независимо от типа набора данных[12].

Мелкие СНН требуют большого количества нейронов в FC-слоях, а также большего количества FC-слоев для более широких наборов данных по сравнению с более глубокими наборами данных и наоборот[12].

Более глубокие СНН работают лучше, чем мелкие модели на более глубоких наборах данных. Напротив, мелкие архитектуры работают лучше, чем более глубокие архитектуры для более широких наборов данных. Эти наблюдения могут помочь при принятии решения о выборе архитектуры глубокого/мелкого СНН[12].

**Выходной слой**

Количество нейронов в этом слое задается в соответствии с требуемыми условиями. Если требуется классификация, то количество нейронов обычно соотносится с количеством категорий, подлежащих классификации[8].

## Параметры нейронных сетей

**1.7.1 Функция активации**

Функция активации вводит нелинейность в нейронную сеть, имитирующую функцию, при которой только нейронные электрические сигналы, превышающие определенный порог, могут быть переданы следующему нейрону[4]. Без этой нелинейности не было бы большого преимущества в соединении нескольких слоев друг с другом. Как правило, функция активации применяется индивидуально к каждому элементу его входного вектора[9].

Сверточные нейронные сети используют различные функции активации для описания сложных объектов. Аналогично функции нейронной модели человеческого мозга, функция активации здесь представляет собой блок, который определяет, какая информация должна быть передана следующему нейрону. Каждый нейрон в нейронной сети принимает выходное значение нейронов из предыдущего слоя в качестве входных данных и передает обработанное значение следующему слою. В многослойной нейронной сети существует функция между двумя слоями[9]. Структура функции активации, её структура представлена на рисунке 6.

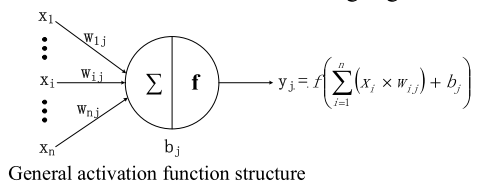


Рисунок 6 - Структура функции активации

На рисунке представляет входной признак; n признаков одновременно вводятся в нейрон j; представляет значение веса связи между входным признаком и нейроном j; представляет внутреннее состояние нейрона j, которое является значением смещения; и - выходной сигнал нейрона j. 𝑓(∙) - функция активации, которая может быть сигмовидной функцией, функцией *tanh(x)*, выпрямленной линейной единицей(ReLU)[9].

Если функция активации не используется или используется линейная функция, входные данные каждого слоя будут линейной функцией выходных данных предыдущего слоя. В этом случае независимо от того, сколько слоев имеет нейронная сеть, выходные данные всегда представляют собой линейную комбинацию входных данных, что означает, что скрытые слои не имеют никакого эффекта. Эта ситуация и есть примитивный персептрон, который обладает ограниченной способностью к обучению. По этой причине нелинейные функции вводятся как функции активации.[4]

**Сигмовидная функция активации**

Сигмовидная функция является одной из наиболее типичных нелинейных функций активации с общим S-образной формы. При приближении значения к 0, градиент становится более крутым. Сигмовидная функция принимает значения в интервале (0, 1), поэтому ее можно использовать для задач двоичной классификации.[4]

**Тангенсальная функции активации**

Тангенсальная функция принимает значения в интервале (-1, 1). Поскольку среднее значение выходного сигнала *tanh* равно 0, он может достичь своего рода нормализации. Это облегчает освоение следующего уровня.[4]

**Функция активации ReLU**

Выпрямленный линейный блок является еще одной эффективной функцией активации. Когда его функциональное значение равно 0; когда , его функциональное значение равно самому . По сравнению с сигмовидной функцией и функцией *tanh*, значительным преимуществом использования функции ReLU является то, что она может ускорить обучение. Сигмоид и *tanh* участвуют в экспоненциальных операциях, которые требуют деления при вычислении производных, тогда как производная ReLU является константой. Более того, в сигмовидной и тангенциальной функциях, если значение x слишком велико или слишком мало, градиент функции довольно мал, что может привести к медленной сходимости функции. Однако, при , производная ReLU равна 0, а при , производная равна 1, поэтому можно получить идеальный эффект сходимости. AlexNet, лучшая модель в ILSVRC- 2012, использует ReLU в качестве функции активации модели на основе СНН, которая устраняет проблему исчезновения градиента, когда сеть глубокая, и проверяет, что использование ReLU превосходит sigmoid в глубоких сетях.[4]

ReLU не учитывает верхний предел. На практике возможно установить верхний предел, например ReLU6.[4]

Однако, при , градиент ReLU равен 0, что означает, что ошибка обратного распространения будет умножена на 0, в результате чего ошибка не будет передана предыдущему слою. В этом сценарии нейроны будут рассматриваться как инактивированные или мертвые. Поэтому предлагаются некоторые улучшенные версии. Негерметичный ReLU (Leaky ReLU) может уменьшить инактивацию нейронов. При , выходной сигнал негерметичного ReLU равен вместо нуля, где ‘a’ - это фиксированный параметр в диапазоне (1, +∞).[4]

Другим вариантом ReLU является PReLU. В отличие от Leaky ReLU, наклон отрицательной части PReLU основан на данных, а не на предопределенном значении.[4]

**Функция экспоненциальных линейных единиц (ELU)**

Улучшенная версия ReLU. Поскольку ReLU активирован неотрицательно, среднее значение его выходного сигнала больше 0. Эта проблема приведет к смещению следующего блока слоя. Функция ELU имеет отрицательное значение, поэтому среднее значение ее выходного сигнала близкого к 0, что делает скорость сходимости быстрее, чем ReLU. Однако отрицательная часть - это кривая, которая требует множества сложных производных.

**Влияние функции активации**

Линейная функция активации действительно приводит к наихудшей производительности. Поэтому при построении глубокой нейронной сети (более одного слоя) нужно добавить нелинейную функцию. Если нет, то несколько слоев, теоретически, равны одному слою.[4]

Среди этих функций активации скорость конвергенции сигмовидной функции самая низкая. Обычно конечная производительность сигмовидной функции не такая высокая. В результате, если необходима быстрая конвергенция, sigmoid - не лучшее решение.[4]

С точки зрения точности, ELU обладает наилучшей точностью, но лишь немного лучше, чем ReLU, Leaky ReLU и PReLU. Что касается времени обучения, ELU, как правило, занимает больше времени, чем ReLU и Leaky ReLU.[4]

ReLU и Leaky ReLU обладают лучшей стабильностью во время тренировки, чем PReLU и ELU.[4]

Функции активации Tanh, PReLU и ELU с большей вероятностью приведут к колебаниям в конце тренировки.[4]

При обучении глубокой модели СНН с предварительно обученными весами трудно сходиться с помощью функций активации sigmoid и tanh.[4]

Модели, обученные Leaky ReLU и ELU, имеют лучшую точность, чем другие в экспериментах. Но иногда ELU может привести к тому, что сети ничему не научатся. Leaky ReLU обладает лучшими характеристиками с точки зрения точности и скорости обучения.[4]

На рисунке 7 представлены графики некоторых функций активации

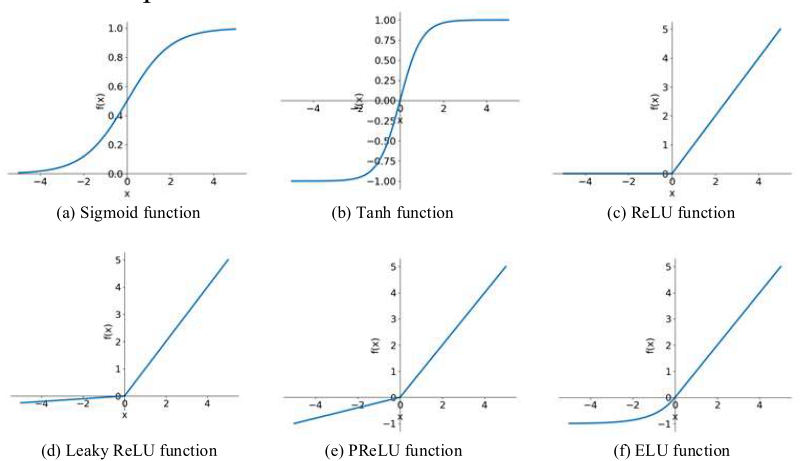


Рисунок 7 - Графики функций активаций[4]

**Правила выбора функции активации**

Для задач двоичной классификации последний уровень может использовать сигмоид; для задач с несколькими классификациями последний уровень может использовать softmax.[4]

Сигмовидные и тангенсальная функции иногда следует избегать из-за исчезновения градиента. Обычно, в скрытых слоях, ReLU или Дырявый ReLU - хороший выбор.[4]

По умолчанию лучше использовать ReLU или Leaky ReLU.[4]

Если в процессе обучения инактивируется много нейронов, необходимо использовать Leaky ReLU, PReLU.[4]

Отрицательный наклон в Leaky ReLU можно установить равным 0,02, чтобы ускорить обучение.[4]

**1.7.2 Функция потерь**

Функция потерь или функция затрат используется для вычисления расстояния между прогнозируемым значением и фактическим значением. Функция потерь обычно используется в качестве критерия обучения в задаче оптимизации. Функция потерь может быть использована со сверточными нейронными сетями для решения задач регрессии и классификации, целью которых является минимизация функции потерь. Общие функции потерь включают Среднее абсолютное Ошибка (MAE), Среднеквадратичная ошибка (MSE), Перекрестная энтропия и т.д.[4]

**Функция потерь для задач классификации**

В сверточных нейронных сетях, когда дело доходит до задач классификации, приходится обрабатывать множество функций потерь.[4]

Наиболее типичный из них, называемый потерей перекрестной энтропии, используется для оценки разницы между распределением вероятностей, полученным в результате текущего обучения, и фактическим распределением. Эта функция сравнивает прогнозируемую вероятность с фактическим выходным значением (0 или 1) в каждом классе и вычисляет штрафное значение на основе расстояния от них. Штраф является логарифмическим, поэтому функция выдает меньший балл (0,1 или 0,2) для меньших различий и больший балл (0,9 или 1,0) для больших различий.[4]

Потеря перекрестной энтропии также называется потерей softmax, что указывает на то, что она всегда используется в СНН со слоем softmax. Например, AlexNet, Inception v1 и ResNet используют кросс-энтропийные потери в качестве функции потерь.[4]

Другим вариантом перекрестной энтропии является потеря softmax с большой маржой. Целью этого также является внутриклассовое сжатие и межклассовое разделение. Потеря softmax с большой маржой добавляет разницу между различными классами и вводит регулярность маржи через угол матрицы весовых ограничений.[4]

**Правила выбора функции потерь**

При использовании моделей СНН для решения проблем регрессии выбрать потери L1 или потери L2 в качестве функции потерь.[4]

При решении задач классификации выбрать остальные функции потерь.[4]

Перекрестная потеря энтропии - самый популярный выбор, обычно появляющийся в моделях СНН со слоем softmax в конце.[4]

Выбор функции потерь в СНН также зависит от сценария применения. Например, для распознавания лиц, в наиболее часто используются контрастные потери и триплетные потери.[4]

**1.7.3 Оптимизатор**

В СНН часто приходится оптимизировать невыпуклые функции. Математические методы требуют огромных вычислительных мощностей, поэтому в процессе обучения используются оптимизаторы для минимизации функции потерь для получения оптимальных параметров сети за приемлемое время. Распространенными алгоритмами оптимизации являются Momentum, RMSProp, Adam и др.[4]

Существует три вида методов градиентного спуска, которые могут использоваться для обучения наших моделей СНН: Пакетный градиентный спуск (BGD), Стохастический градиентный спуск (SGD) и Мини-пакетный градиент Спуск (MBGD).

BGD указывает, что для получения градиента для каждого обновления необходимо рассчитать целый пакет данных, который может обеспечить сходимость к глобальному оптимуму выпуклой плоскости и локальному оптимуму невыпуклой плоскости. Однако использование BGD довольно медленно, поскольку необходимо рассчитать средний градиент для всей партии образцов. Кроме того, это может быть сложно для данных, которые не подходят для вычисления в памяти. Следовательно, BGD практически не используется при обучении моделей на основе СНН на практике.[4]

Напротив, SGD использует только один образец для каждого обновления. Очевидно, что время SGD для каждого обновления значительно меньше, чем BGD, поскольку для вычисления требуется только один градиент выборки. В этом случае SGD подходит для онлайн-обучения. Однако SGD быстро обновляется с высокой дисперсией, что приведет к серьезным колебаниям целевой функции. С одной стороны, колебание вычисления может привести к тому, что вычисление градиента выйдет за пределы локального оптимума и, наконец, достигнет лучшей точки; с другой стороны, SGD может никогда не сходиться из-за бесконечных колебаний.[4]

На основе BGD и SGD был предложен MBGD, который сочетает в себе преимущества BGD и SGD. MBGD использует небольшую партию выборок для каждого обновления, так что он может не только выполнять более эффективный градиентный анализ, чем BGD, но и уменьшать дисперсию, делая сходимость более стабильной.[4]

Среди этих трех методов MBGD является наиболее популярным. Многие классические модели СНН используют его для обучения своих сетей в оригинальных статьях, таких как AlexNet, VGG, Inception v2, ResNet и DenseNet. Он также был использован в FaceNet, DeepID и DeepID2.[4]

Другим часто используемым оптимизатором является Adaptive Moment Оценка (Adam). Adam хранит как среднее значение экспоненциального спада прошлых квадратных градиентов, как алгоритм Adadelta, так и среднее значение экспоненциального спада прошлых градиентов, как алгоритм Momentum. Практика доказала, что алгоритм Adam хорошо работает во многих задачах и применим ко многим различным структурам сверточных нейронных сетей.[4]

**Правила выбора оптимизатора**

Мини-пакет (*mini-batch*) следует использовать для того, чтобы найти компромисс между вычислительными затратами и точностью каждого обновления.[4]

Если возникают чрезмерные колебания или расхождения, хорошим выбором может быть снижение скорости обучения.

**Выводы**

Выбор специфичной для набора данных архитектуры СНН, который в основном выполняется либо с помощью опыта, либо с помощью специальных знаний, является трудоемким и подверженным ошибкам процессом.

## Методы построения ансамбля

Первоначально успех ансамблевых методов был теоретически исследован в регрессионных задачах. Исследователи доказали, что правильный ансамблевый классификатор гарантирует меньшую квадратическую ошибку по сравнению с отдельными предикторами классификатора.[13]

Стратегии ансамбля в целом классифицируются следующим образом:[13]

**1.8.1 Бэггинг**

Основная идея пакетирования состоит в том, чтобы сгенерировать серию независимых наблюдений с тем же размером и распределением, что и исходные данные. *Бэггинг* увеличивается два шага в оригинальных моделях: Во-первых, генерация упакованных выборок и передача каждого пакета выборок базовым моделям, а во-вторых, стратегия объединения прогнозов нескольких предикторов. Объединение выходных данных базовых предикторов может варьироваться, поскольку для задач классификации используется в основном голосование большинством голосов, в то время как стратегия усреднения используется в задачах регрессии для получения выходных данных ансамбля.[13]

Хотя объединение может привести к увеличению вычислительной сложности, но *беггинг* обладает тем свойством, что оно может быть распараллелено и может привести к эффективному сокращению времени обучения при условии наличия аппаратного обеспечения для запуска параллельных моделей. Поскольку модели глубокого обучения имеют большое время обучения, следовательно, оптимизация нескольких глубоких моделей на разных учебных пакетах не является возможным вариантом.[13]

**1.8.2 Бустинг**

Метод усиления используется в моделях ансамбля для преобразования слабой модели обучения в модель обучения с лучшим обобщением. Такие методы, как голосование большинством голосов в случае проблем классификации или линейная комбинация слабых учащихся в задачах регрессии, приводят к лучшему прогнозированию по сравнению с одним слабым учащимся.[13]

Известные методы *boosting* AdaBoost и Gradient Boosting.[13]

**1.8.3 Стратегия слияния решения**

Ансамблевое обучение обучает нескольких базовых учащихся и объединяет результаты базовых учащихся, используя некоторые правила. Правило, используемое для объединения выходных данных, определяет эффективную производительность ансамбля. Большинство моделей ансамбля фокусируются на архитектуре ансамбля, за которой следует их простое усреднение для прогнозирования результата ансамбля.

**Невзвешенное усреднение модели**

Невзвешенное усреднение результатов базовых учащихся в ансамбле является наиболее распространенным подходом к объединению решений. Результаты базовых учащихся усредняются, чтобы получить окончательный прогноз модели ансамбля. Архитектуры глубокого обучения имеют высокую дисперсию и низкое смещение, таким образом, простое усреднение моделей ансамбля улучшает производительность обобщения за счет уменьшения дисперсии между моделями. Усреднение базовых учащихся выполняется либо непосредственно по результатам базовых учащихся, либо по прогнозируемым вероятностям классов с помощью функции softmax[13]:

(4)

где - вероятностный результат i -го блока для j -го базового ученика, - результат i -го блока для j -го базового ученика, а K - количество классов.

Невзвешенное усреднение является оптимальным выбором, когда успеваемость базовых учащихся сопоставима.[13]

**Голосование большинства**

Подобно невзвешенному усреднению, голосование большинством голосов объединяет результаты базовых учащихся. Однако вместо того, чтобы брать среднее значение вероятностных результатов, голосование большинством голосов подсчитывает голоса базовых учащихся и предсказывает окончательные метки как метки с большинством голосов. По сравнению с невзвешенным усреднением голосование большинством голосов менее предвзято в отношении результатов конкретного базового учащегося, поскольку эффект смягчается подсчетом голосов большинством голосов.[13]

**Выводы**

Применение различных подходов ансамблирования приводит к повышению производительности и повышению качества предсказания.

Однако существует проблема выбора подходящих моделей СНН для комбинирования их в ансамбле. До сих пор не определены критерии выбора моделей в архитектуре ансамблей.

1. Разработка ансамбля нейронных сетей

## Описание используемого набора изображений

Обучение и тестирование модели осуществляется на датасете COVID-19 Radiography Dataset[14], который собирается и обновляется группой исследователей из Катарского университета, Доха, Катар, и Университета Дакки, Бангладеш, вместе исследователями из Пакистана и Малайзии в сотрудничестве с врачами. На момент исследования база данных содержит 21165 рентгеновских снимков грудной клетки для случаев COVID-19, а также изображений нормальной и вирусной пневмонии. Этот набор данных по COVID-19, нормальным и другим легочным инфекциям.

**Структура данных.**

Датасет состоит из четырех частей.

1. Изображения, маркированные COVID:

* 3615 изображений собрано из различных доступных публично баз данных, онлайн ресурсов, и опубликованных исследований:
* 2473 изображений из PadChest датасета. Большого набора данных рентгеновских снимков грудной клетки с аннотированными отчетами с несколькими метками.
* 183 изображений из Немецкой медицинской школы
* 559 изображений из SIRM, Github, Kaggle & Tweeter
* 400 изображений из других ресурсов Github.

1. Обычные изображения:

* 10192 изображения из двух датасетов:
* 8851 Радиологического общества Северной Америки (RSNA)
* 1341 Kaggle.

1. Изображения непрозрачности легких:

* 6012 Изображения CXR непрозрачности легких собраны из набора данных CXR Радиологического общества Северной Америки (RSNA).

1. Изображения вирусной пневмонии:

* 1345 Данные о вирусной пневмонии собраны из базы данных рентгеновских снимков грудной клетки (пневмония).

Изображения сохранены в формате «png». Размер изображений 299х299 пикселей. Весь набор изображений размечен метками – «COVID», «Normal», «Pneumonia», «Lung\_Opacity».

Датасет загружен на сетевой диск Google Drive, который подключен к проекту в среде Google Colaboratory. Управление подключением/отключением сетевого диска к проекту осуществляется с помощью модуля drive из google.colab.

## Техническая реализация

Программная реализация ансамбля нейронных сетей выполнена на языке Python, имеющего реализацию широкого набора готовых модулей для обработки данных и реализаций многих архитектур нейронных сетей через соответствующие библиотеки.

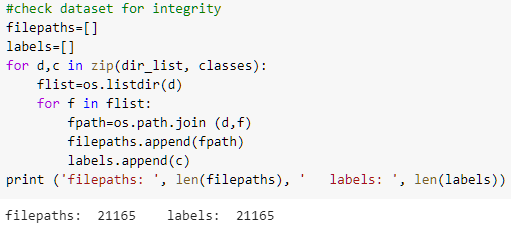
Используются библиотеки Tensorflow и Keras и другие вспомогательные библиотеки обработки данных и вычислений.

Вычисления производились на удаленном сервере Google Colaboratory.

## Структура программного обеспечения

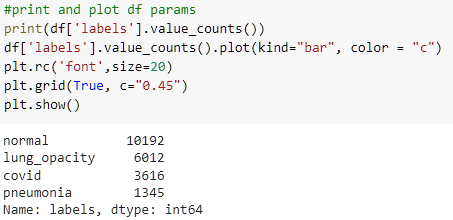
**Предобработка данных**

В листинге 1 представлен код анализа датасета. В выводе функции указывается количество файлов с изображениями и количество маркеров. Равенство значений говорит о том, что все изображения имеют маркер, определяющий класс изображения, - таким образом, датасет полностью размечен.



Листинг 1 - Подсчет файлов и маркеров в датасете

Определение распределения изображений по классам представлено в листинге 2. На рисунке 8 представлена гистограмма распределения изображений по классам.



Листинг 2 - Подсчет количества изображений каждого класса

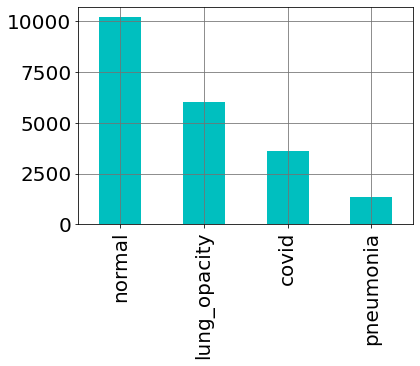
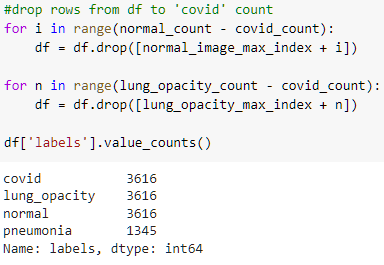


Рисунок 8 - Гистограмма распределения изображений по классам

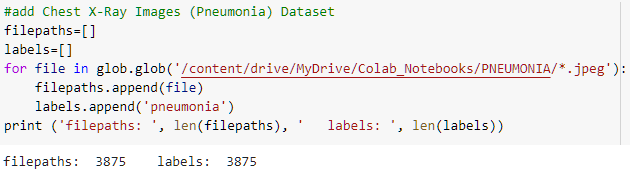
Для балансировки датасета по классам необходимо получить равное количество изображений каждого вида. Поскольку обучение направлено на определение изображения класса «covid», то количество изображений других классов необходимо привести к количеству изображений целевого класса.

В листинге 3 представлен код удаления из датасета изображений классов, с большим представлением в наборе данных – «normal» и «lung\_opacity».



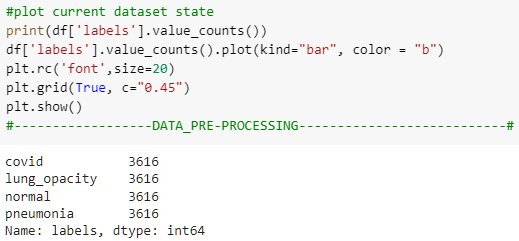
Листинг 3 - Выравнивание по классам

Для выравнивания количества изображений класса «pneumonia» используется датасет [Chest X-Ray Images (Pneumonia)](https://www.kaggle.com/code/vexxingbanana/96-val-acc-covid-19-radiography/data)[15], содержащий 3875 файлов с изображением данного класса (листинг 4).



Листинг 4 - Состав датасета [Chest X-Ray Images (Pneumonia)](https://www.kaggle.com/code/vexxingbanana/96-val-acc-covid-19-radiography/data)

После добавления необходимого количества файлов датасет становится сбалансированным. Структура сбалансированного набора изображений представлена на рисунке 9, вывод числа изображений по классам представлен в листинге 5.



Листинг 5 - Подсчет числа изображений по классам

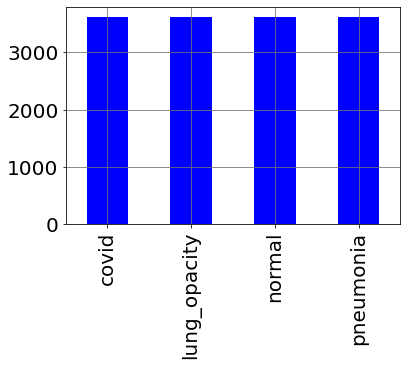


Рисунок 9 - Распределение изображений по классам в сбалансированном датасете

Для обучения и тестирования модели датасет разделен на тренировочную и тестовую части в пропорции 80% - тренировочная, 20% - тестовая.

Каждая сеть из ансамбля обучается на тренировочной части датасета. Применено случайное перемешивание порядка расположения изображений в датасете.

На рисунке 10 представлен порядок расположения файлов изображения в тестовом (test\_df) и тренировочном (train\_df) датасетах. Подсчет числа файлов представлен в листинге 6.

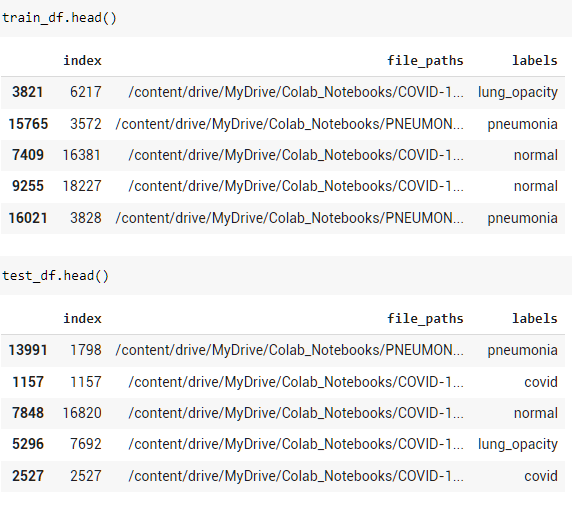
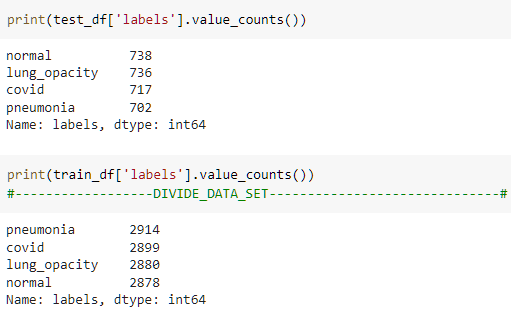


Рисунок 10 - Смешанный порядок расположения файлов в датасете



Листинг 6 - Подсчет количества файлов в датасетах

**Обучение нейронных сетей ансамбля**

Для всех индивидуальных моделей установлены идентичные параметры обучения, основываясь на описании правильного выбора параметров для нейронной сети, приведенном в параграфе 1.7.

Функция активации ReLU

Последний уровень softmax

Функция потерь – кросс энтропия

Оптимизатор –Адам

Количество эпох - ?(25)

Коллбэки с терпением 10 эпох?

Лернинг рейт = 0,0001

Для

– Train the models - DenseNet201, ResNet50V2, and Inceptionv3 using training set images and do the loss minimization based on the test set images. Calculate the weights of the 5-fold cross validation based on the test set.

– Run the trained models on the validation set images and select class label value 0 or 1 based on weighted average ensembling of the 3 models.

The overall proposed approach, as summarized in Fig. 6, includes:

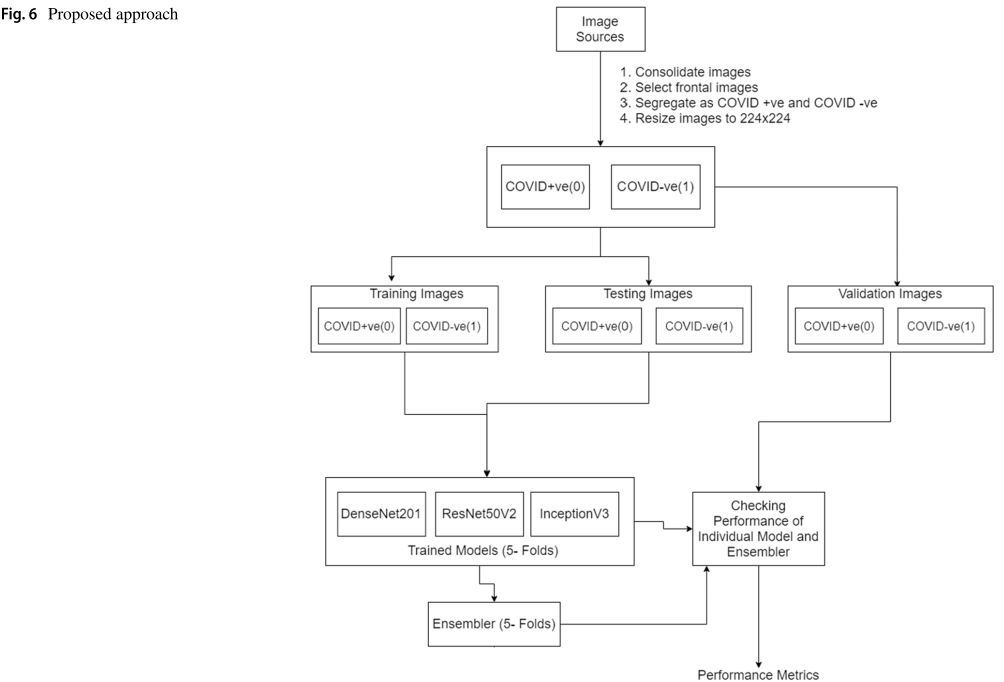
– Consolidate CXR images for healthy subjects, patients having pneumonia or other bacterial infection and COVID patients from different sources.

– Retain only frontal CXR images.

– Resize images to a uniform size.

– Divide the images into three portions—training, testing and validation datasets. One small portion is retained as validation set to test the efficacy of the trained model while the remaining portion is divided into 5 folds. Each time one separate fold is picked up as test data and the remaining folds as training data.

– While dividing the images into training, testing and validation sets, ensure that there is no patient overlap i.e., different images of the same patient is not present in multiple sets.



Выбор функций активаций

Выбор нейронных сетей

Предоработка

Метрики качества

Сравнение с бенчмарком

Для сравнения качества модели стоит сравнить с показателями отдельных сетей, составляющих ансамбль.

Сколько эпох обучения было выбрано.

Какой оптимизатор был выбран

Какие другие параметры обучения были выбраны.

Какое время в эпохах заняло обучение?

Матрицы смешения (конфьюза)

1. Результаты исследования

1.8 Описание ансамбля моделей нейронных сетей.

Ансамбль нейронных сетей, исследуемый в этой работе, состоит из моделей с небольшим числом параметров.

Показать архитектуры выбранных сетей.

Inception V3

Xception

ResNet50V2

Методика эксперимента

Результат эксперимента

Сравнение и оценка результата

1. Экономическое обоснование

В выпускной квалификационной работе описывается алгоритм настройки нейронных сетей для решения задачи классификации биомедицинских изображений, а также метод расчета выходного значения модели.

Существующие аналоги настроены под определенные характеристики данных, для которых они реализованы. Отличием описанного в работе метода является использование несложных нейронных сетей для ускорения получения результата и использование набора изображений из нескольких открытых баз данных.

В ходе выполнения работы были решены задачи сбора и обработки изображений для анализа, выбор архитектуры используемых моделей нейронных сетей, выбор технологии программирования разрабатываемой модели.

**Определение продолжительности проведения работ**

Расчет затрат на написание программы основан на плане работ.

Данные по продолжительности каждого этапа работ представлены в таблице 1.

Продолжительность работ определяется расчетным путем с помощью экспертных оценок по формуле:

, (5)

где – ожидаемая длительность j-й работы; и – наименьшая и наибольшая, по мнению эксперта, длительность работы.

Таблица 1 – Длительность этапов работы

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование работы | Длительность работы, человеко-дни. | | |
|  |  |  |
| Разработка ТЗ | 3 | 5 | 3,8 |
| Анализ ТЗ и работа с базами данных | 3 | 5 | 3,8 |
| Определение технологи программирования | 2 | 3 | 2,4 |
| Выбор платформы запуска программы | 2 | 3 | 2,4 |
| Работа с программой для разработки модели | 18 | 30 | 22,8 |
| Настройка параметров модели | 3 | 8 | 5 |
| Оформление пояснительной записки | 15 | 20 | 17 |
| ИТОГО |  |  | 57,2 |

**Оценка величины заработной платы и социальных отчислений участников работ**

Для каждого исполнителя определена ставка заработной платы за день. Ставка заработной платы за день определяется исходя из месячной заработной платы соответствующего исполнителя. Дневная ставка заработной платы определяется делением заработной платы (оклад) за месяц на количество рабочих дней в месяце (21 рабочий день).

Для студента в качестве месячной заработной платы принимается заработная плата инженера. По данным Управления Федеральной службы статистики по г. Санкт-Петербургу и Ленинградской области (ПЕТРОСТАТ), в Санкт-Петербурге средняя заработная плата инженера в 2021 году составила 101 600 рублей[21]. Зарплату руководителя составляет 140 000 рублей.

Ставка в день составляет 4838 и 6666 рублей для будущего инженера и руководителя соответственно.

Данные по стоимости этапов работ представлены в таблице 2.

Таблица 2 - Стоимость работ

|  |  |  |  |
| --- | --- | --- | --- |
| Этапы и содержание выполняемых работ | Исполнитель | Трудоемкость, , чел.-дни. | Ставка, руб./день |
| Разработка ТЗ  Анализ ТЗ и работа с базами данных | Руководитель | 1,8 | 6666 |
| Студент | 2 | 4838 |
| Определение технологи программирования | Студент | 3,8 | 4838 |
| Выбор платформы запуска программы | Студент | 2,4 | 4838 |
| Работа с программой для разработки модели | Студент | 2,4 | 4838 |
| Настройка параметров модели | Руководитель | 2 | 6666 |
| Студент | 20,8 | 4838 |
| Разработка ТЗ | Студент | 5 | 4838 |
| Анализ ТЗ и работа с базами данных | Студент | 17 | 4838 |

На основе данных о трудоемкости выполняемых работ и ставки за день соответствующих исполнителей рассчитываются расходы на заработную плату исполнителей и отчислений на страховые взносы на обязательное социальное, пенсионное и медицинское страхование.

Расходы на основную заработную плату исполнителей определяются по формуле:

, (6)

где - расходы на основную заработную плату исполнителей (руб.); k – количество исполнителей; - время, затраченное *i*-м исполнителем на проведение исследования (дни); - ставка *i*-го исполнителя (руб./день).

руб.

Расходы на дополнительную заработную плату исполнителей определяются по формуле:

, (7)

где - расходы на дополнительную заработную плату исполнителей (руб.); - расходы на основную заработную плату исполнителей (руб.); - норматив дополнительной заработной платы 14 (%).

руб.

Отчисления на страховые взносы на обязательное социальное, пенсионное и медицинское страхование с основной и дополнительной заработной платы исполнителей определяются по формуле:

, (8)

где - отчисления на социальные нужды с заработной платы (руб.); - расходы на основную заработную плату исполнителей (руб.); - расходы на дополнительную заработную плату исполнителей (руб.); - норматив отчислений на страховые взносы на обязательное социальное, пенсионное и медицинское страхование (30%).

руб.

**Оценка затрат, связанных с приобретением необходимых комплектующих**

Модель реализована по технологии бесплатного облочного сервиса для машинного обучения. Используются открытые библиотеки, свободно распространяемый код.

**Расчёт затрат на содержание и эксплуатацию оборудования**

Затраты на содержание и эксплуатацию оборудования определяются из расчета на 1 час работы оборудования с учетом стоимости и производительности оборудования:

, (9)

где – затраты на содержание и эксплуатацию оборудования (руб.); – расчетная себестоимость одного машино-часа работы оборудования на *i*-й

технологической операции (руб./м-ч)1; – количество машино-часов, затрачиваемых на выполнение *i*-й технологической операции (м-ч).

При выполнении работы использовался персональный компьютер, себестоимость одного машино-часа работы которого составляет 5,76 руб./м-ч. (или 46,1 руб./м-д.). Время использования персонального компьютера – 60 дней.

Затраты на содержание и эксплуатацию составят:

руб.

**Определение величины амортизационных отчислений используемых основных средств**

Амортизационные отчисления по основному средству (персональный компьютер) за год определяются как:

, (10)

где – амортизационные отчисления за год по i-му основному средству (руб.); – первоначальная стоимость i-го основного средства (руб.); – годовая норма амортизации *i*-го основного средства (%).

 рассчитывается по формуле:

, (11)

где – первоначальная стоимость основного средства (руб.); – ликвидационная стоимость оборудования (руб.) (5% от первоначальной стоимости оборудования); – нормативный срок службы основного средства (лет).

Годовая норма амортизации:

.

Амортизационные отчисления по основному средству:

руб.

Величина амортизационных отчислений по основному средству, используемому при работе над ВКР, определяется по формуле:

, (12)

где – амортизационные отчисления по *i*-му основному средству, используемому в работе над ВКР (руб.); – амортизационные отчисления за год по *i*-му основному средству (руб.); , – время, в течение которого используется основное средство (мес.).

Таким образом, величина амортизационных отчислений по основному средству, используемому при работе над ВКР:

руб.

**Расходы на услуги сторонних организаций**

Для доступа к удаленным вычислительным серверам использовался доступ в интернет. Услуга связи предоставляется сторонней организацией. Стоимость услуги составляет 450 рублей в месяц, из них НДС по ставке 20% составляет 75 рублей.

Стоимость услуг без НДС составляет 375 рублей в месяц.

С учетом продолжительности выполнения исследования – 2 месяца, суммарно стоимость услуг связи составляет 750 рублей

**Определение величины накладных расходов**

В статью «Накладные расходы» включаются расходы на управление и хозяйственное обслуживание.

Величина накладных расходов составляет 25% от фонда заработной платы:

руб.

**Расчёт совокупной величины затрат.**

Для оценки итоговой величины затрат, связанных с выполнением работы, все проведенные расчеты представлены в таблице 3.

Таблица 3 - Смета затрат на ВКР

|  |  |
| --- | --- |
| Наименование статьи | Сумма, руб. |
| Расходы на оплату труда | 323395,2 |
| Отчисления на социальные нужды |  |
| Расходы на услуги сторонних организаций | 750 |
| Расходы на содержание и эксплуатацию оборудования |  |
| Амортизационные отчисления |  |
| Накладные расходы |  |
| ИТОГО затрат | 324145,2 |

**Вывод**

По итогам расчетов совокупные затраты составили 324145,2 руб. Большая часть затрат относится к расходам на оплату труда и отчислениям на социальные нужды.

Стоимость разработки алгоритма анализа изображения соответствует рыночным условиям.

Внедрение программного обеспечения в медицинские организации может сократить время диагностики, снизить уровень ошибки, повысить качество предоставления медицинской услуги, и более оперативно поставить диагноз и назначить соответствующее лечение.

# Заключение

В работе описаны примеры развития архитектур сверточных нейронных сетей для решения задачи классификации рентгеновских снимков грудного отдела человека. Дано подробное описание слоев, составляющих нейронные сети, функций активации, потерь, оптимизатора. Указаны рекомендации по выбору архитектуры и настройке параметров, в зависимости от типа задачи и характеристик данных.

Дано описание методов составления ансамблей нейронных сетей.

Целью работы является разработка ансамбля нейронных сетей для обучения на рентгеновских снимках для распознавания кодвида-19.

Проведена проверка на качество выбранных параметров сети на независимых наборах данных.

В результате три типа нейронных сетей были адаптированы для выработки единого результата. Модель показала точность классификации …

Применение разработанной модели позволит предоставить врачам подкрепленный анализ изображения за небольшой промежуток времени, что повысит скорость и эффективность принятия решения, и следовательно, качество медицинского обслуживания.

# Список используемых источников

1. Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. Scientiﬁc Reports 10(1), 19549 (Nov 2020). [Электронный ресурс]. *URL: https://www.nature.com/articles/s41598-020-76550-z.pdf* (дата обращения: 20.04.2022).
2. An Efficient CNN Model for COVID-19 Disease Detection Basedon X-Ray Image Classification [Электронный ресурс]. *URL:https://www.researchgate.net/publication/351683364\_An\_Efficient\_CNN\_Model\_for\_COVID-19\_Disease\_Detection\_Based\_on\_X-Ray\_Image\_Classification* (дата обращения: 28.04.2022).
3. AC-CovidNet: Attention Guided Contrastive CNN for Recognition of Covid-19 in Chest X-Ray Images [Электронный ресурс]. *URL:* [*https://arxiv.org/abs/2105.10239*](https://arxiv.org/abs/2105.10239) (дата обращения: 18.04.2022).
4. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects Zewen Li, Wenjie Yang, Shouheng Peng, Fan Liu, Member, IEEE [Электронный ресурс]. *URL:https://www.researchgate.net/publication/340475800\_A\_Survey\_of\_Convolutional\_Neural\_Networks\_Analysis\_Applications\_and\_Prospects* (дата обращения: 28.05.2022).
5. Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network [Электронный ресурс]. *URL:* [*https://arxiv.org/pdf/2003.11617.pdf*](https://arxiv.org/pdf/2003.11617.pdf) (дата обращения: 20.04.2022).
6. «Artificial Intelligence for Health and Health Care» December 2017 [Электронный ресурс] *URL: https://www.healthit.gov/sites/default/files/jsr-17-task-002\_aiforhealthandhealthcare12122017.pdf* (дата обращения: 07.05.2022).
7. «Рентгенологи приступили к сборке всероссийского датасета для проверки сервисов ИИ.» Цифровая экономика 07-10-2020. [Электронный ресурс]. *URL: https://www.comnews.ru/digital-economy/content/209439/2020-10-07/2020-w41/rentgenologi-pristupili-k-sborke-vserossiyskogo-dataseta-dlya-proverki-servisov-ii* (дата обращения: 20.04.2022).
8. Blind Channel Identification Aided Generalized Automatic Modulation Recognition based on Deep Learning [Электронный ресурс]. *URL: https://www.researchgate.net/publication/335086346\_Blind\_Channel\_Identification\_Aided\_Generalized\_Automatic\_Modulation\_Recognition\_Based\_on\_Deep\_Learning/figures?lo=1* (дата обращения: 28.04.2022).
9. An Introduction to Deep Learning for the Physical Layer Tim O’Shea, Senior Member, IEEE, and Jakob Hoydis, Member, IEEE [Электронный ресурс]. *URL: https://arxiv.org/pdf/1702.00832.pdf* (дата обращения: 28.05.2022).
10. Pooling Methods in Deep Neural Networks, a Review [Электронный ресурс]. *URL:* *https://www.researchgate.net/publication/344277235\_Pooling\_Methods\_in\_Deep\_Neural\_Networks\_a\_Review* (дата обращения: 28.05.2022).
11. A review on modern defect detection models using DCNNs – Deep convolutional neural networks Andrei-Alexandru Tulbure, Adrian-Alexandru Tulbure, Eva-Henrietta Dulfa [Электронный ресурс]. *URL: https://www.researchgate.net/publication/350617050\_A\_review\_on\_modern\_defect\_detection\_models\_using\_DCNNs\_-Deep\_convolutional\_neural\_networks* (дата обращения: 26.05.2022).
12. Impact of fully connected layers on performance of convolutional neural networks for image classiﬁcation, Neurocomputing, [Электронный ресурс]. *URL:* [*https://arxiv.org/abs/1902.02771*](https://arxiv.org/abs/1902.02771) (дата обращения: 28.05.2022).
13. Ensemble deep learning: A review [Электронный ресурс]. *URL:* [*http://arxiv.org/abs/2104.02395v2*](http://arxiv.org/abs/2104.02395v2) (дата обращения: 28.04.2022).
14. COVID-19 Radiography Dataset [Электронный ресурс]. *URL: https://www.kaggle.com/datasets/preetviradiya/covid19-radiography-dataset* (дата обращения: 05.05.2022).
15. [Chest X-Ray Images (Pneumonia)](https://www.kaggle.com/code/vexxingbanana/96-val-acc-covid-19-radiography/data) [Электронный ресурс]. *URL: https://md-datasets-public-files-prod.s3.eu-west-1.amazonaws.com/31ab5ede-ed34-46d4-b1bf-c63d70411497* (дата обращения: 02.05.2022).
16. [Электронный ресурс]. *URL: https://www.tensorflow.org/api\_docs/python* (дата обращения: 07.05.2022).
17. ПЕТРОСТАТ [Электронный ресурс]. *URL: https://petrostat.gks.ru/storage/mediabank/* (дата обращения: 02.06.2022)