

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт кибернетики и информационных технологий

Кафедра кибербезопасность, обработка и хранение информации

Арманкызы Рената

Поиск аномалий в рентгеновских снимках при помощи глубокого машинного обучения

ДИПЛОМНАЯ РАБОТА

Специальность 5В070300 – Информационные системы

Алматы 2021

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт кибернетики и информационных технологий

Кафедра кибербезопасность, обработка и хранение информации

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой
КБОиХИ

канд. техн. наук, ассистент-
профессор

 Н. А. Сейлова

« 28 »

05 2021г.

ДИПЛОМНАЯ РАБОТА

На тему: Поиск патологий в рентгеновских снимках при помощи глубокого
машинного обучения

Специальность 5В070300 – Информационные системы

Выполнил: Арманкызы Р.

Научный руководитель
магистр технических наук,
лектор

 Аристомбаева М.Т.

« 28 » 05 2021г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт кибернетики и информационных технологий

Кафедра кибербезопасность, обработка и хранение информации

5B070300 – Информационные системы

УТВЕРЖДАЮ

Заведующий кафедрой КБОиХО
канд. техн. наук, ассистент-
профессор

 Н. А. Сейлова

« 28 » 05 2021 г.

ЗАДАНИЕ

на выполнение дипломной работы

Обучающемуся: Арманкызы Рената

Тема: Поиск патологий в рентгеновских снимках при помощи глубокого машинного обучения

Утверждена приказом Ректора Университета № 2131-б от 24.11.2020г.

Срок сдачи законченной работы 02.06.2020г.

Исходные данные к дипломному проекту: результаты преддипломной практики, открытые источники с обширной базой данных для обучения методов глубокого обучения, сбор теоретического материала.

Краткое содержание дипломной работы:

- а) Общая характеристика заболеваний легких и способы их диагностирования;
- б) Анализ предметной области;
- в) Информационное и программное обеспечение;
- г) Результаты исследования.

Рекомендуемая основная литература: *из 12 наименований*

ГРАФИК

подготовки дипломной работы (проекта)

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Общая характеристика заболеваний легких и способы их диагностирования	15.02.2020г.	
Анализ предметной области	20.04.2020г.	
Информационное и программное обеспечение	15.05.2020г.	
Результаты исследования	20.05.2020г.	

Подписи

консультантов и нормоконтролера на законченную дипломную работу (проект) с указанием относящихся к ним разделов работы (проекта)

Наименование разделов	Консультанты, Ф.И.О. (уч.степень, звание)	Дата подписания	Подпись
Поиск патологий в рентгеновских снимках при помощи глубокого машинного обучения	Аристомбаева М.Т., Лектор, магистр технических наук		
Нормоконтролер	Кабдуллин М.А., тьютор		
Программная часть			

Научный руководитель: _____ Аристомбаева М.Т.

Задание принял к исполнению обучающийся: _____ Арманкызы Р.

Дата

"__" _____ 202

АННОТАЦИЯ

Дипломдық жобаның тақырыбы: «Терең машиналық оқыту арқылы рентген суреттеріндегі ауытқуларды іздеу».

Дипломдық жұмыстың көлемі 38 парақты құрайды, оның ішінде 20 сурет және 1 кесте. Түйінді сөздер: машиналық оқыту, тереңдетіп оқыту, CNN, рентгенография.

Осы дипломдық жұмыстың негізгі міндеттері - терең машиналық оқытудың қолданыстағы әдістерін медициналық проблемаларға қолдану аясында және пневмониямен ауыратын науқастардың суреттерімен жоғары ұқсастығын ескере отырып, Covid19 инфекциясын кеуде қуысының рентгенограммасында анықтау қабілетін зерттеу.

Жұмыс барысында медицинада қолданылатын машиналық оқыту әдістерінің әр түрлі топологияларын, сондай-ақ төменгі тыныс алу жолдарының аурулары бойынша медициналық көрсеткіштерді қамтитын тақырыптық аймақты талдау арқылы ақпараттық жүйені құру бойынша шаралар қабылданды, CNN әдісінің жиырма вариациясы арқылы нейрондық жүйе құру және ең жақсы әдісті анықтау.

Болашақта өнімділікті арттыру және осы ақпараттық жүйені пайдалануды жеңілдететін веб-қосымша құру арқылы жобаны жетілдіру жоспарлануда.

АННОТАЦИЯ

Тема дипломного проекта: “Поиск аномалий в рентгеновских снимках при помощи глубокого машинного обучения”.

Объем дипломной работы 38 страницы, в том числе 20 рисунок и 1 таблица. Ключевые слова: машинное обучение, глубокое обучение, CNN, рентгенография.

Основными целями данной дипломной работы являются изучение существующих методов глубокого машинного обучения в контексте применения к медицинским задачам и возможность идентифицировать инфекцию Covid19 по рентгеновским снимкам грудной клетки, учитывая высокое сходство снимков со снимками пациентов имеющие пневмонию.

В ходе работы были проведены мероприятия по составлению информационной системы путём анализа предметной области, который включает как разные топологии методов машинного обучения, применяемые в медицине так же и медицинские показания по заболеваниям нижних дыхательных путей, построение нейронных сетей по двадцати вариациям метода CNN и выявление наилучшего метода.

Во последующем намечается усовершенствование проекта путем повышения производительности и созданием web-приложения для облегчения использования данной информационной системы.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1. Общая характеристика заболеваний легких и способы их диагностирования	7
1.1 Пневмония	7
1.2 COVID-19	8
2. Анализ предметной области	10
2.1 Области применения машинного и глубокого обучения	11
2.2 Постановка задачи	12
2.3 Мотивация и вклад	13
3. Информационное и программное обеспечение	13
3.1 Используемые базы данных	14
3.2 CNN	14
3.2.1 Сверточный слой	15
3.2.2 Объединяющий слой	16
3.2.3 Полносвязный слой	16
3.2.4 Предварительно обученные модели	17
3.3 Экспериментальная установка	20
3.3 Показатели производительности	20
4. Результаты исследования	22
4.1 Подготовка данных	22
4.2 Результаты обучения	25
4.3 Использование предложенного метода	28
Заключение	29
Список использованной литературы	30
ПРИЛОЖЕНИЕ А	31

ВВЕДЕНИЕ

Ситуация, вызванная новым коронавирусом, изменила поведение каждого человека по отношению ко всему. Мы вынуждены пребывать на карантине дома, выполнять работу и учиться из дома, ходить на улице в масках и всегда носить с собой дезинфицирующие средства, каждый человек теперь стал вынужден адаптироваться к так называемой серии карантинных мероприятий. Сложность выявления инфекции Covid19 на начальной стадии связана с большим сходством ее симптомов с инфекцией, вызванной пневмонией. В связи с этим вирус достаточно быстро распространилась по всему миру. По этой причине диагностика легочных заболеваний стало самой актуальной задачей не только для работников медицинской отрасли, но и для всего населения нашей планеты. Так же актуальность и практический аспект данных проблем связаны не только с коронавирусной пандемией, но и из-за роста отрасли промышленности с неудовлетворительным контролем выделения вредоносных веществ в воздух, которые повышают предрасположенность к таким заболеваниям как рак легких, туберкулез, пневмония и другие. Поэтому появляется необходимость в постоянном наблюдении состояния легких для предупреждения либо обнаружения заболеваний, до того, как оно нанесёт суровый ущерб здоровью.

Следовательно, насущная проблема в данный момент — это быстрое обнаружение; и это становится все более важным по мере того система здравоохранения оказывается перегруженной потоком данных о пациентах. Необходимость создания автоматизированного компьютеризированного процесса становится все более очевидной. Имея это в виду, предлагается использовать радиомные признаки визуализации, используя для этой цели глубокое обучение.

Глубокое обучение влечет за собой обучение на основе необработанных данных для автоматического обнаружения представлений, необходимых для обнаружения или классификации. В контексте медицинских изображений он напрямую использует значения пикселей изображений (вместо извлеченных или выбранных функций) на входе; таким образом, устраняются ручные ошибки, вызванные неточной сегментацией или последующим извлечением признаков. Сверточные нейронные сети (CNN) представляют собой одну из популярных моделей глубокого обучения. Прорыв в CNN произошел с соревнованием ImageNet в 2012 году, где частота ошибок при распознавании объектов сократилась почти вдвое.

Ожидается, что алгоритмы искусственного интеллекта, наряду с клиническими и радиологическими, функциями, полученными на основе рентгеновских снимков грудной клетки, окажут огромную помощь в проведении масштабных программ обнаружения, которые могут иметь место

в любой стране, имеющей доступ к рентгеновскому оборудованию и помощь в эффективной диагностике COVID-19.

В этом сценарии машинное обучение (ML) и глубокое обучение (DL) предлагают быстрые, автоматизированные, эффективные стратегии для обнаружения отклонений и извлечения ключа особенности измененной паренхимы легких, которые могут быть связаны с конкретными сигнатурами вируса COVID-19.

Рентгенография — технически простой и наиболее доступный для населения метод предварительной диагностики заболевания, в том числе отслеживания процесса протекания заболевания. Медицинские специалисты в процессе определения образов в изображениях при принятии решений сталкиваются с рядом вопросов: неполная и неточная начальная информация; большая изменчивость атрибутов и маленькие размеры подборки;

Актуальность темы дипломной работы связана со значительным распространением исследуемого явления и заключается в необходимости разработки рекомендаций по совершенствованию работы в рассматриваемой области.

- Объект исследования – патологии легких
- Предмет исследования – рентгенограмма, разделенная по категориям: здоровые, пациенты больные пневмонией, пациенты больные covid19.

Целью дипломной работы является выявить самый эффективный подход к обнаружению аномалий, которые могут быть показателями пневмонии или covid19 по рентгеновским снимкам грудной клетки.

1. Общая характеристика заболеваний легких и способы их диагностирования

1.1 Пневмония

Легкие - жизненно важный орган в физиологической структуре человека, и заболевания легких могут серьезно повлиять на состояние здоровья.

Пневмония — это распространенная легочная инфекция, при которой воспаляются воздушные мешки легких. Эти мешочки также могут заполняться жидкостью, гноем и клеточными остатками. Это может быть вызвано вирусами, грибами или бактериями. В клиниках пневмонию диагностируют при самых разных заболеваниях используя методы визуализации, таких как рентген грудной клетки, КТ и МРТ.

Бактериальная пневмония может быть легкой или тяжелой. Тяжесть пневмонии зависит от:

- сила бактерий;
- как быстро поставят диагноз и начнут лечение;
- возраст пациента;
- общее самочувствие;
- если у пациента есть другие заболевания.

Наиболее частые симптомы бактериальной пневмонии:

- кашель с густой желтой, зеленой или кровянистой слизью;
- колющая боль в груди, которая усиливается при кашле или дыхании;
- внезапное начало озноба, достаточно сильного, чтобы вызвать сильный дрожь во всем теле;
- лихорадка 38-40 ° С или выше (лихорадка ниже 38° С у пожилых людей).

Пожилые люди будут разделять все симптомы с более молодыми людьми, но гораздо чаще испытывают замешательство и головокружение. У пожилых людей также может быть меньшая температура.

Наиболее характерной рентгенологической картиной пневмонии вызванной бактерией может быть (Рисунок 1.1):

- Очаги затемнения разных размеров;
- Очаги затемнения имеют не четкие контуры;
- Очаги затемнения не гомогенной структуры.

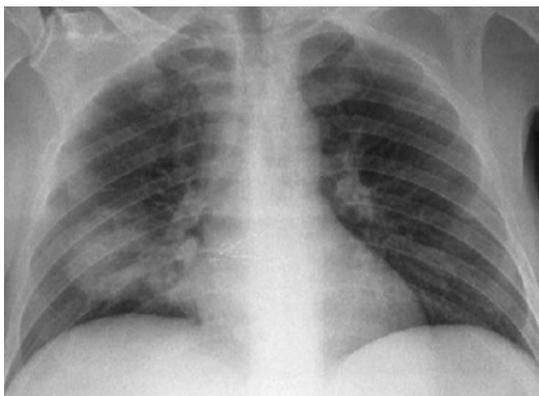


Рисунок 1.1 - Округлая пневмония: уплотнение в правой нижней доле легкое этого взрослого пациента. *Streptococcus pneumoniae* был определен в посевах мокроты.

1.2 COVID-19

Коронавирус (CoV) принадлежит к большому семейству вирусов, вызывающих заболевания, варьирующиеся от обычного гриппа до более тяжелых проявлений, таких как ближневосточный респираторный синдром (MERS-CoV) и тяжелый острый респираторный синдром (SARS-CoV). Новый коронавирус (nCoV) или COVID-19 — это новый штамм под названием SARS-CoV2, ранее не обнаруженный у людей. Хотя эта вспышка началась как эпидемия в Ухане, Китай, сегодня она уже серьезно затронула все страны мира как пандемия. В настоящее время не существует эффективного лекарства от этого вируса, и существует острая необходимость в расширении глобальных знаний о его механизмах инфицирования, распределении повреждений легких не только для обнаружения болезней или диагностики, но и для поддержки дизайна лечебной терапии.

Это семейство вирусов носит название коронавируса из-за наличия на поверхности клетки шиповидной короны, которую можно увидеть на Рисунке 1.2.

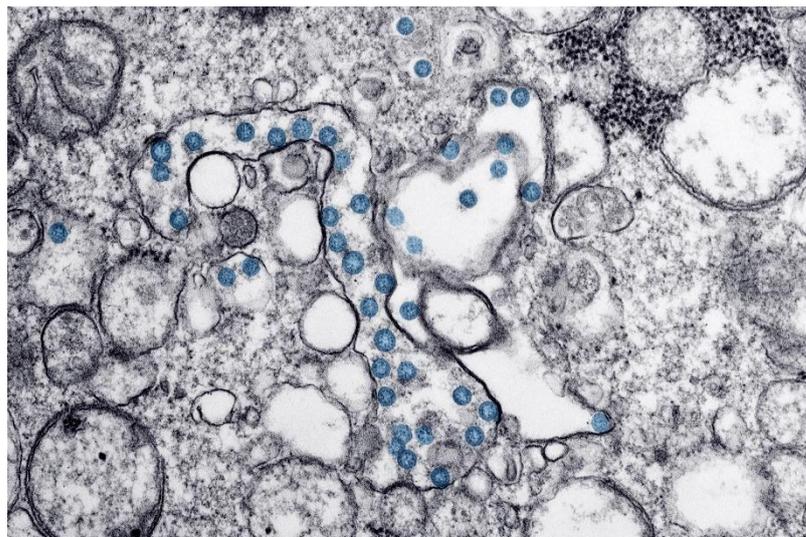


Рисунок 1.2 - Изображение COVID-19 сделанное просвечивающим электронным микроскопом.

Наиболее частые симптомы COVID-19:

- Жар;
- Сухой кашель;
- Усталость;
- Боль в груди;

К другим симптомам, которые встречаются реже и могут повлиять на некоторых пациентов, относятся:

- Потеря вкуса или запаха;
- Заложенность носа;
- Больное горло;
- Головная боль;
- Боль в мышцах или суставах;
- Озноб или головокружение.

Респираторное заболевание, связанное с коронавирусом, обычно клинически проявляется как пневмония с преобладающими результатами визуализации атипичной или организуемой пневмонии. Обычная рентгенография очень полезна для оценки заболевания COVID-19 и последующего наблюдения. Это дает точное представление о течении болезни.

При интерстициальной пневмонии, то есть имеющей вирусную этиологию, в частности вызванной коронавирусом, могут быть следующие признаки при рентгенографии:

- Затемнение легочных полей диффузное, чаще двухстороннее;
- Диффузные изменения по типу матового стекла;

- Перибронхильное затемнение;
- Склерозирование(утолщение) междолевых перегородок;
- Рост перибронхиальных узлов, в дальнейшем есть риск развития бронхоэктазов;
- Усиленный сосудистый рисунок, особенно в области корней легкого;
- При высокой степени фиброза отмечается уменьшение объема легких, преимущественно в нижних отделах.



Рисунок 1.3 – Развитие вирусной пневмонии

На рисунке 1.3 можно увидеть рентгенограмму 41-летней женщины с положительным COVID-19. Первоначальный рентген грудной клетки (A) был нормальным. На 3-й день контрольная рентгенограмма грудной клетки (B) показывает четко выраженный большой одиночный легочный узел (стрелка) на периферии левой средней зоны легкого с TSS равным 1. На 18-й день (C) поражение частично рассосалось со слабой плотностью (стрелка)

2. Анализ предметной области

2.1 Области применения машинного и глубокого обучения

Машинное обучение и глубокое обучение — это разновидности искусственного интеллекта. Глубокое обучение — это развитая и продвинутая фаза машинного обучения. В машинном обучении создаются алгоритмы, которые учатся на данных и проводят анализ.

Глубокое обучение отличается от машинного обучения тем, что оно работает в искусственной нейронной сети, которая точно представляет человеческий мозг. Та же сеть позволяет машинам анализировать данные так же, как это делают люди. Такие машины со способностью к глубокому обучению не должны действовать в соответствии с инструкциями программистов.

Глубокое обучение стало возможным благодаря огромным объемам данных, которые мы создаем и потребляем ежедневно. Каждая модель глубокого обучения широко использует данные для облегчения обработки данных.

Машинное обучение и глубокое обучение помогли человечеству улучшить и ускорить не только многие профессиональные и производственные процессы, а также и повседневную жизнь.

В настоящее время они используются во многих областях и отраслях:

- Компьютерное зрение. Глубокие нейронные сети обеспечивают новейшее обнаружение объектов, классификацию изображений, восстановление изображений и сегментацию изображений. Настолько, что они даже обеспечивают распознавание рукописных цифр в компьютерной системе, стало обычным делом.
- Боты на основе глубокого обучения. Исследователи Nvidia разработали систему искусственного интеллекта, которая помогает роботам учиться на демонстративных действиях человека. Роботы-хозяйки, которые выполняют действия на основе входов искусственного интеллекта из нескольких источников, довольно распространены. Подобно тому, как человеческий мозг обрабатывает действия на основе прошлого опыта и сенсорных входов, инфраструктуры глубокого обучения помогают роботам выполнять задачи в зависимости от различных мнений ИИ.
- Автоматические переводы. Автоматические переводы существовали до того, как было добавлено глубокое обучение. Но глубокое обучение помогает машинам выполнять улучшенные переводы с гарантированной точностью, которой не хватало в прошлом. Кроме того, глубокое обучение также помогает в переводе текста, полученном из

изображений - что-то совершенно новое, что было бы невозможно при традиционной интерпретации на основе текста.

- Опыт работы с клиентами. Многие компании уже используют машинное обучение для работы с клиентами. К жизнеспособным примерам относятся онлайн-платформы самообслуживания. Кроме того, многие организации теперь полагаются на глубокое обучение для создания надежных рабочих процессов. Большинство из нас уже знакомы с использованием чат-ботов организациями. По мере того, как это применение глубокого взгляда становится более зрелым, мы можем ожидать дальнейших улучшений в этой области.
- Автономные транспортные средства. Миллионы моделей искусственного интеллекта могут информировать одну машину во время движения по дороге. Многие считают, что вождение автомобиля на базе ИИ безопаснее, чем вождение человека.
- Анализ изображений и создание подписей. Одним из величайших достижений глубокого обучения является способность идентифицировать изображения и создавать для них интеллектуальные подписи.
- Генерация текста. Теперь машины могут создавать новый текст с нуля. Они могут выучить пунктуацию, грамматику и стиль текста и написать эффективные новостные статьи. Робо-журналисты, использующие модели глубокого обучения, готовят точные отчеты о матчах уже как минимум три года. И навыки не ограничиваются исключительно составлением отчетов о матчах.

2.2 Постановка задачи

Основная цель данной работы – выявить самый эффективный подход к обнаружению аномалий, которые могут быть показателями пневмонии или covid19 по рентгеновским снимкам грудной клетки. Для этого, необходимо решить следующие задачи:

- Изучение существующих методов глубокого машинного обучения и применение их к медицинским задачам;
- Возможность идентифицировать инфекцию Covid19 по рентгеновским снимкам грудной клетки, учитывая высокое сходство снимков со снимками пациентов имеющие пневмонию;
- Проанализировать решение о комбинациях различных методов глубокого обучения, дающих более эффективную модель идентификации Covid19;

- Оценка производительности классификаторов машинного обучения для лучшей идентификации Covid19;
- Предложение наилучшего набора классификаторов машинного обучения для более эффективного Covid19.

2.3 Мотивация и вклад

Ожидается, что искусственный интеллект (ИИ) применительно к радиомным особенностям с помощью визуализации грудной клетки, рентгеновского снимка, наряду с другими клиническими, патологическими и геномными параметрами, окажет полезную поддержку в этом направлении.

Другая основная цель - уменьшить ошибку обобщения при достижении более точного диагноза.

Вклады резюмируются следующим образом:

- Глубокое сверточное представление признаков используется для извлечения высоко репрезентативных признаков с использованием современных глубоких дескрипторов CNN. Используемый подход способен отличить COVID-19 и бактериальную пневмонию от здоровых людей с помощью рентгеновского снимка грудной клетки или изображения компьютерной томографии и, следовательно, обеспечивают более высокую точность и высокую скорость по сравнению с ручными способами.
- Решить проблему переобучения в глубоком обучении из-за ограниченного количества итераций.
- Чтобы повысить способность к обобщению, а также уменьшить предвзятость модели к набору данных не применяются никакие дополнительные данные или обширные методы предварительной обработки.
- Предлагаемый подход значительно сокращает время обнаружения при достижении удовлетворительной точности, что является превосходным преимуществом для разработки в реальном или близком к реальному времени делая заключения по клиническим применениям.

3. Информационное и программное обеспечение

3.1 Используемые базы данных

В этой работе представлены рентгеновские снимки грудной клетки 341 пациентов с COVID-19. Они были получены из репозитория GitHub с открытым исходным кодом от доктора Джозефа Коэна. Этот репозиторий состоящие из рентгеновских снимков грудной клетки принадлежат преимущественно пациентам с острым респираторным дистресс-синдромом (ОРДС), COVID-19, пневмония. Кроме того, рентгенограммы грудной клетки 2772 бактериальных и 1493 вирусных пневмонии были использованы из репозитория Kaggle под названием «Chest X-Ray Images (Pneumonia)».

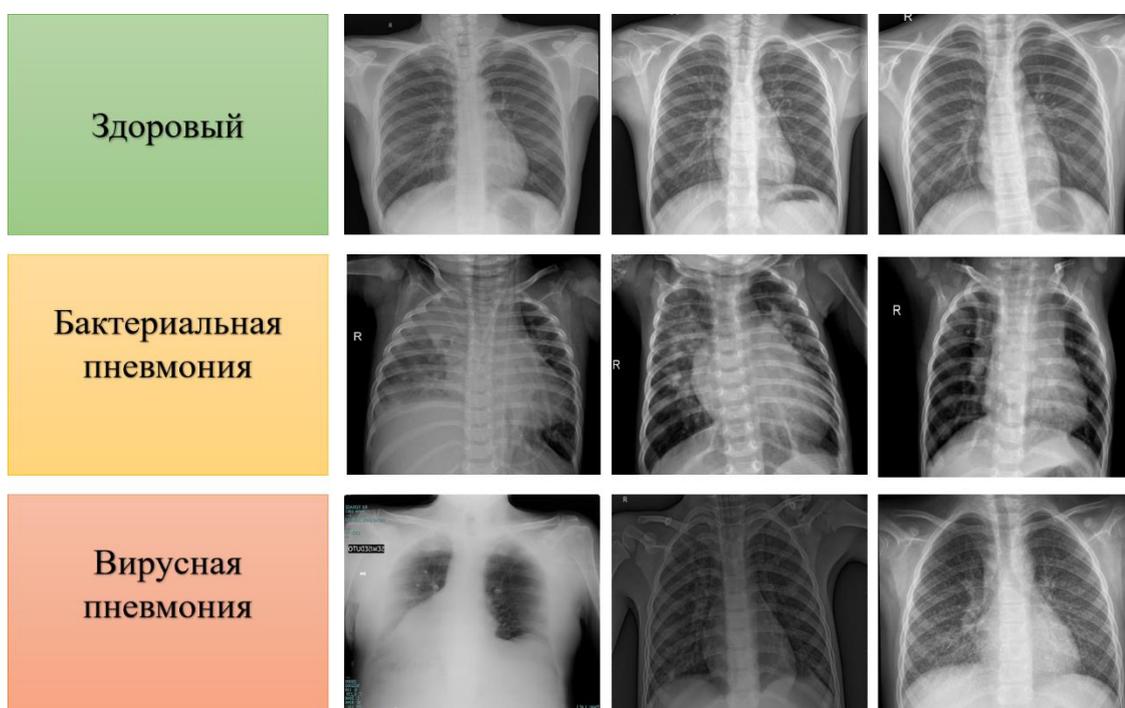


Рисунок 3.1 – Рентгеновские снимки при разных заболеваниях.

3.2 CNN

Сверточная нейронная сеть (CNN) — это класс глубоких нейронных сетей, который чаще всего используется в задачах распознавания изображений. Что касается того, как работает CNN, изображения, представленные как ввод должны быть распознаны машиной и преобразованы в формат, который можно обрабатывать. По этой причине изображения первым делом преобразовывается в матричный формат. Система определяет какое изображение принадлежит какому классу на основе различий в изображениях и, следовательно, в матрицах. Он изучает влияние этих

различий во время обучения фазы, а затем делает прогнозы для новых изображений, используя их. CNN состоит из трех различных слоев: сверточного, объединённого и полносвязного, чтобы выполнять эти операции эффективно. Процесс извлечения признаков происходит как в сверточном, так и в объединяющем слоях. С другой стороны, процесс классификации происходит на полносвязном уровне. Эти слои последовательно исследуются ниже.

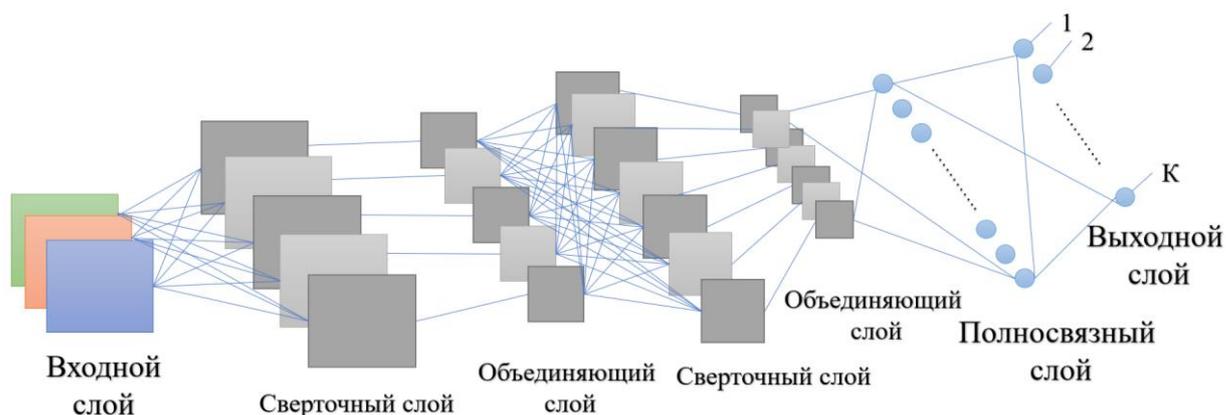


Рисунок 3.2 – Архитектура CNN.

3.2.1 Сверточный слой

Сверточный слой — это базовый уровень CNN. Он отвечает за определение особенностей рисунка. В этом слое входное изображение пропускается через фильтр(ядро). Этот слой применяет несколько ядер, которые скользят по изображению для извлечения в паттерн низко- и высокоуровневых характеристик. Во время «скольжения» ядро умножает между собой значения пикселей и затем суммирует по ниже предоставленной формуле. Ядро представляет собой матрицу размером 3×3 или 5×5 , которая должна быть преобразована с помощью матрицы входного шаблона. Параметр шага - количество шагов, настроенных для перехода по входной матрице. Выход сверточного слоя можно представить как:

$$x_j^l = f \left(\sum_{a=0}^N w_j^{l-1} * y_a^{l-1} + b_j^l \right)$$

где, x_j^l это j-я карта признаков в слое l, w_j^{l-1} – j-ое по счету ядро в слое l-1, y_a^{l-1} это a-я по счету карта признаков в слое l-1, b_j^l указывает смещение j-й карты признаков в слое l, N – общее количество признаков в слое l – 1, (*) представляет собой процесс векторной свертки.

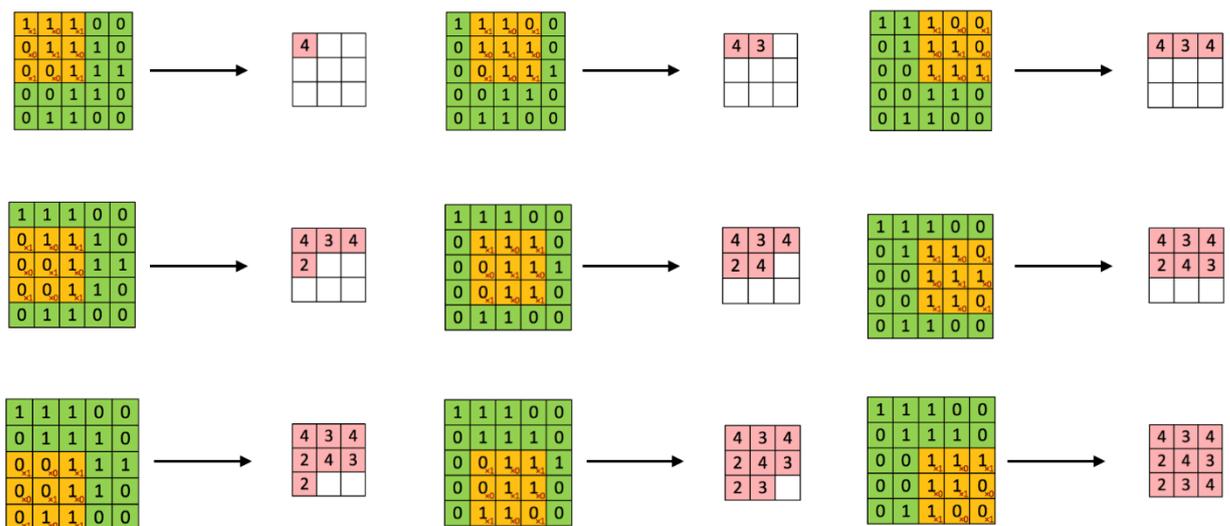


Рисунок 3.3 – Пример сверточного слоя. Зеленым – изображение. Желтым – ядро. Розовым – карта признаков.

3.2.2 Объединяющий слой

Второй слой после сверточного — это объединяющий слой. Слой объединения обычно применяется к созданным картам объектов для уменьшения количества карт объектов и параметров сети путем применения соответствующих математических вычислений. В этой работе использованы максимальный пул и глобальный средний пул. В процессе максимального пула выбираются только максимальное значение, используя размер матрицы, указанный в каждой карте характеристик, что приводит к уменьшению выходных нейронов. Там есть также слой глобального среднего пула, который использовался только до полносвязного слоя, сокращающий данные до одного измерения. Слой глобального среднего пула подключается к полносвязному слою.

3.2.3 Полносвязный слой

Полносвязный слой - последний и самый важный слой CNN. Этот слой функционирует как многослойный перцептрон. Вход для полностью подключенного слоя — это выход из последнего объединенного или сверточного слоя, который сглаживается и затем передается в полносвязный. Этот сглаженный вектор затем соединяется с несколькими полносвязными слоями, которые аналогичны искусственным нейронным сетям и выполняют те же математические операции!

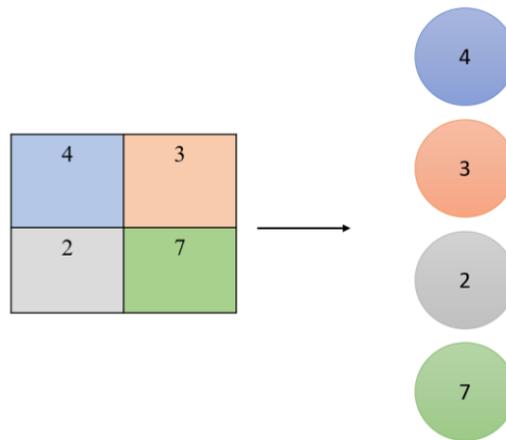


Рисунок 3.4 – Сглаживание карты признаков до матрицы для передачи в полносвязный слой.

3.2.4 Предварительно обученные модели

Обучающие модели сверточной нейронной сети (CNN) созданием миллионов параметров с нуля не только требует очень много времени, но и требует высокопроизводительного оборудования. Чтобы преодолеть эти проблемы, параметры и веса моделей, обученных на разных наборах данных, переносятся в новую модель. Помимо переданных частей, процесс обучения также осуществляется через недавно добавленные слои. Утверждается, что он особенно успешен даже в немногих наборах данных. Кроме того, этот метод позволяет быстрее получать результаты с меньшими затратами на расчет.

При анализе медицинских данных одной из самых больших трудностей, с которыми сталкиваются исследователи, является ограниченное количество доступных наборов данных. Для моделей глубокого обучения часто требуется много данных. Маркировка этих данных экспертами требует больших затрат и времени. Самым большим преимуществом использования метода обучения передачи является то, что он позволяет обучать данные с меньшими затратами времени и сил на вычисления, а так обучаться на меньших наборах данных. С методом трансферного обучения, который широко используется в области глубокого обучения, информация, полученная с помощью предварительно обученной модели на большом наборе данных, передается в модель для обучения.

В этом исследовании были построены 20 моделей на основе CNN, это разные топологии ResNet, Inception, InceptionResnet, VGG и DenseNet.

- Resnet.

Модель остаточной нейронной сети (Residual neural network) — это улучшенная версия CNN.

Чтобы упростить задачу обучения глубоких сетей, новая архитектура поставляется с остаточной структурой обучения, известной как ResNet. В этой

архитектуре переформулированы нейронные сети. Путем обучения с использованием остаточных функций в отношении входных данных слоя.

ResNet также называется остаточной сетью, которая вводит концепцию пропуска соединения для решения проблемы исчезающего градиента. Проблема. Это предотвращает искажение, которое проявляется как сеть и становится глубже и сложнее. Вариант ResNet

ResNet-50 используется как одна из моделей. Архитектура ResNet-50 показана на рисунке 3.5. В ней используется 50-уровневая сеть. и обучен с использованием набора данных ImageNet. Архитектура ResNet-50 развивается сверточным способом.

ResNet101 и ResNet152 состоят из 101 и 152 уровня, соответственно, из-за сложенных друг на друга строительных блоков ResNet. Появилась возможность загрузить предварительно обученную версию более чем миллиона изображений из базы данных ImageNet. В результате сеть научилась богатым функций для широкого диапазона изображений.

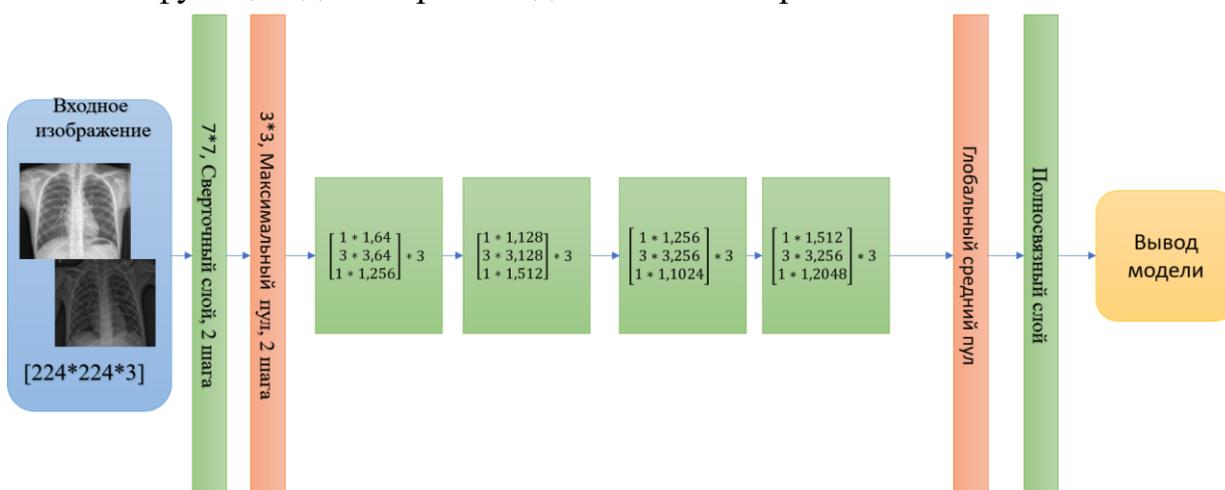


Рисунок 3.5 – Архитектура ResNet.

- InceptionV3

InceptionV3 - это вид сверточной нейронной сети. Модель состоит из базового блока, называемого «Начальная ячейка», в котором мы выполняем серию сверток в разных масштабах и затем агрегируем результаты. Для экономии вычислений используются свертки 1x1 для уменьшения глубины входного канала. Для каждой ячейки мы изучаем набор фильтров 1x1, 3x3 и 5x5, которые могут научиться извлекать из входных данных функции в разных масштабах. Также используется максимальное объединение, хотя и с «одинаковым» заполнением для сохранения размеров, чтобы выходные данные могли быть правильно объединены.

Как и в случае модели ResNet50 сеть обучается с помощью набора данных ImageNet.

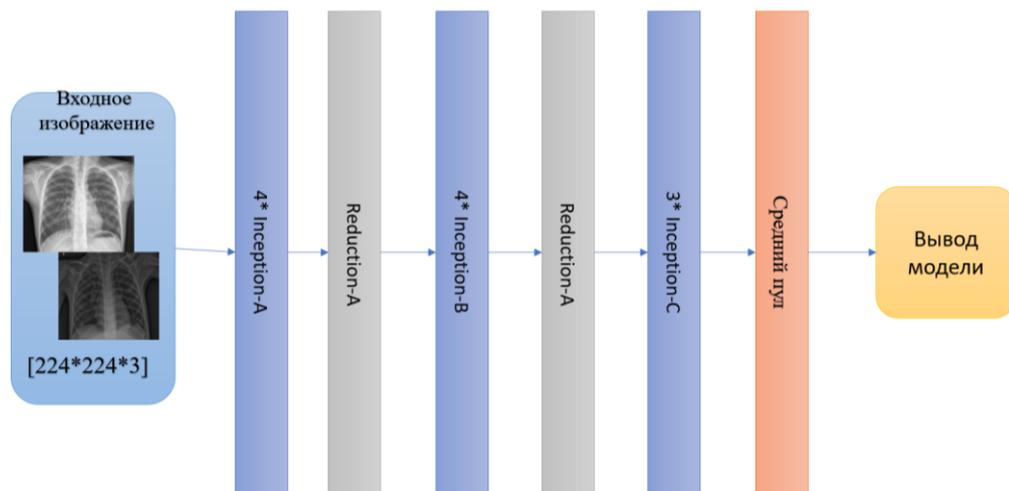


Рисунок 3.6 – Архитектура InceptionV3

- Densenet121

Хотя ResNet изменил способ, которым глубокие нейронные сетевые функции параметризованы, но все еще могут существовать некоторые концептуальные расширения, выполняемое архитектурой DenseNet. Таким образом, эта архитектура является логическим продолжением ResNet.

Архитектура DenseNet121 показана на рисунке 3.7 Как показано на рисунке, DenseNet121 использует сверточные слои, максимальный пул, глобальный средний пул, плотный блок и переходный уровень.

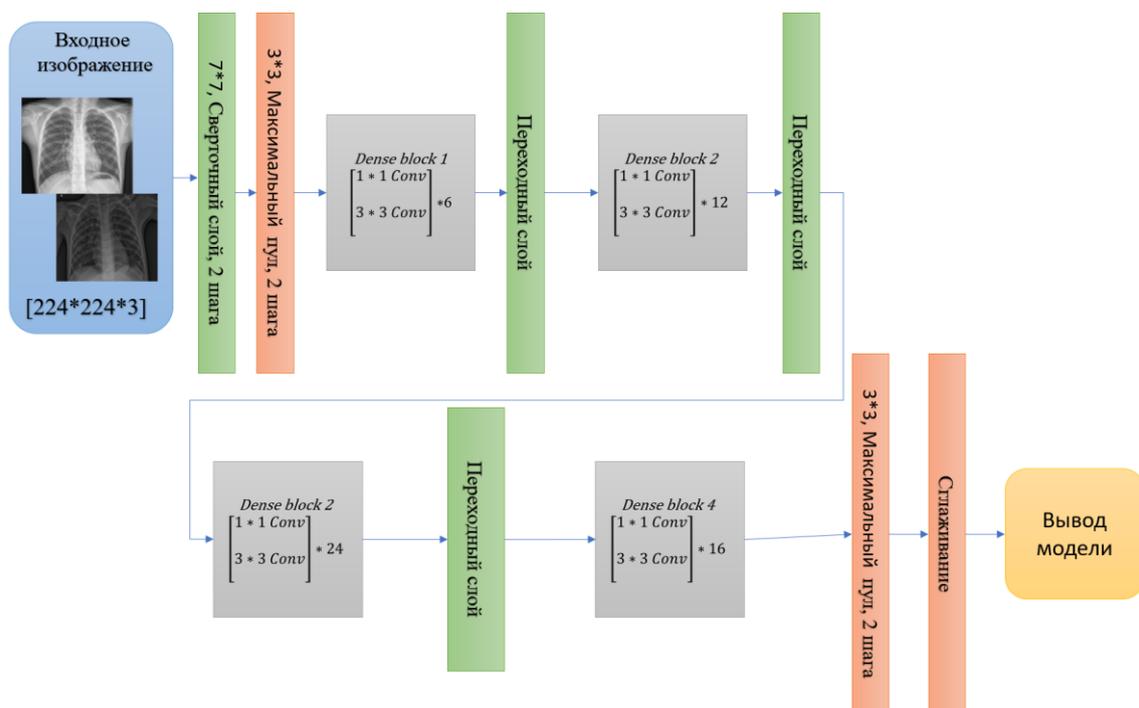


Рисунок 3.7 – Архитектура Densenet121

- EfficientNet

В частности, наш EfficientNet-B4 обеспечивает самую современную точность 84,3% в ImageNet, при этом он в 8,4 раза меньше и в 6,1 раза быстрее при выводе, чем лучший из существующих ConvNet. EfficientNets также хорошо передают данные и достигают высочайшей точности на CIFAR-100 (91,7%), Flowers (98,8%) и 3 других наборах данных для передачи данных с на порядок меньшим количеством параметров.

- VGGNet

Эта сеть отличается своей простотой, в ней используются только сверточные слои 3x3, наложенные друг на друга с нарастающей глубиной. Уменьшение размера обрабатывается путем максимального пула. Затем за двумя полностью связанными слоями, каждый с 4096 узлами, следует классификатор softmax.

3.3 Экспериментальная установка

Для обучения предложенных моделей глубокого трансферного обучения использовался язык программирования Python. Все эксперименты были выполнены бесплатно, онлайн на Linux-сервере Google Colaboratory (Colab) с операционной системой Windows 10 с помощью облачного сервиса с центральным процессором (ЦП), с графическим процессором Tesla K80 Аппаратное (GPU) или тензорного процессора (TPU).

Модели CNN (ResNet50, ResNet101, ResNet152, InceptionV3 и Inception ResNetV2) были предварительно обучены со случайной инициализацией весов путем оптимизации функции кросс-энтропии с помощью оптимизатора адаптивной оценки момента. Размер пакета, скорость обучения и количество эпох были экспериментально установлены равными 3, $1e - 5$ и 30 соответственно для всех экспериментов. Все используемые наборы данных были случайным образом разделены на два независимых набора данных с процентным соотношением 70%, и 20% для обучения и тестирования соответственно.

3.3 Показатели производительности

Для оценки эффективности модели глубокого трансферного обучения применяются следующие пять критериев.

Ассурасу — это соотношение правильно обозначенных предметов ко всему пулу предметов. Точность - самая интуитивная. Точность отвечает на следующий вопрос: сколько диагнозов мы правильно поставили из всех случаев заболеваний?

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Recall — это соотношение правильно обозначенных нашей программой ко всем, кто на самом деле болен. Вспомните ответ на следующий вопрос: скольких из всех больных коронавирусом мы правильно предсказываем?

$$Recall = \frac{TP}{TP + FN}$$

Specificity — это то, что программа правильно помечает всем, кто на самом деле здоров. Специфика отвечает на следующий вопрос: сколько из всех здоровых людей мы правильно предсказали?

$$Specificity = \frac{TN}{TN + FP}$$

Precision — это отношение правильно помеченных программой ко всем помеченным. Precision дает ответ на следующий вопрос: сколько из тех, кого мы назвали коронавирuсно больными, действительно болеют Covid-19?

$$Precision = \frac{TP}{TP + FP}$$

F1-score – это гармоническое среднее (среднее) precision и recall. Оценка F1 является наилучшей, если в системе есть какой-то баланс между точностью (p) и отзывчивостью (r). Напротив, оценка F1 не так высока, если один показатель улучшается за счет другого.

Например, если P равно 1, а R равно 0, оценка F1 равна 0.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

TP (True Positive) – истинно позитивный результат.

TN (True Negative) – истинно негативный результат.

FP (False Positive)– ложно позитивный результат.

FN (False Negative)– ложно отрицательный результат.

4. Результаты исследования

4.1 Подготовка данных

Использованные библиотеки во время работы с данными:

- numpy для работы с многомерными массивами;
- pandas для обработки и анализа данных;
- pathlib для манипуляции директориями;
- matplotlib для визуализации данных;
- tensorflow для облегчения построения и тренировок нейронных сетей;
- seaborn для создания статистических графиков;
- sklearn для работы с машинным обучением.

```
import numpy as np
import pandas as pd
from pathlib import Path
import os.path
import matplotlib.pyplot as plt
import tensorflow as tf
import seaborn as sns
from time import perf_counter
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
from IPython.display import Markdown, display
```

Рисунок 4.1 – Список использованных библиотек

Все полученные изображения из разных источников были загружены, и их общее количество представлено в Рисунке 4.1

Все изображения разделены между собой по диагнозам на:

- NORMAL – Рентгеновский снимок здорового пациента.
- VIRUS – Рентгеновский снимок пациента с вирусной пневмонией.
- PNEUMANIA – Рентгеновский снимок пациента с бактериальной пневмонией.

```
↳ Количество изображений для обучения модели: 5239
   Количество изображений для тестирования модели: 624
   Количество изображений для проверки: 28
```

```
Количество видов диагнозов: 3
```

```
Диагнозы: ['VIRUS' 'PNEUMONIA' 'NORMAL']
```

Рисунок 4.2 – Количество изображений, а также виды маркеров, по которым разделены данные.

Для избежания переобучения данных под разными категориями должно быть примерное одинаковое количество изображений.

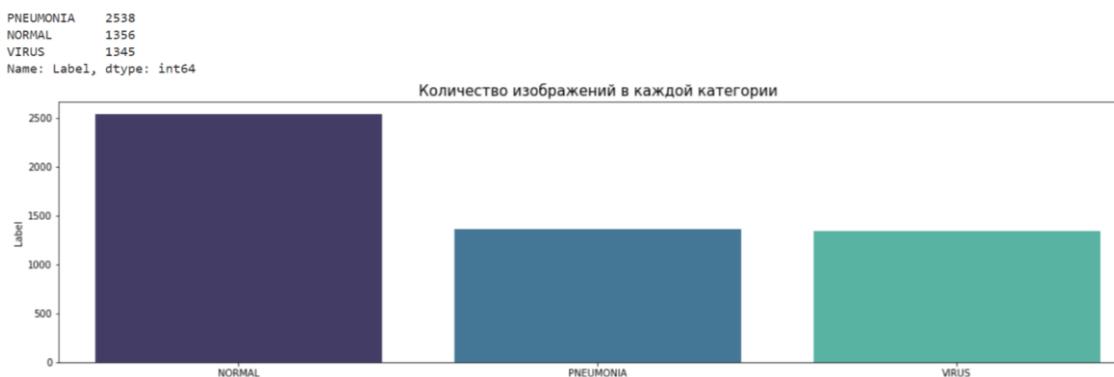


Рисунок 4.2 – Количество и соотношение изображений использованные для обучения.

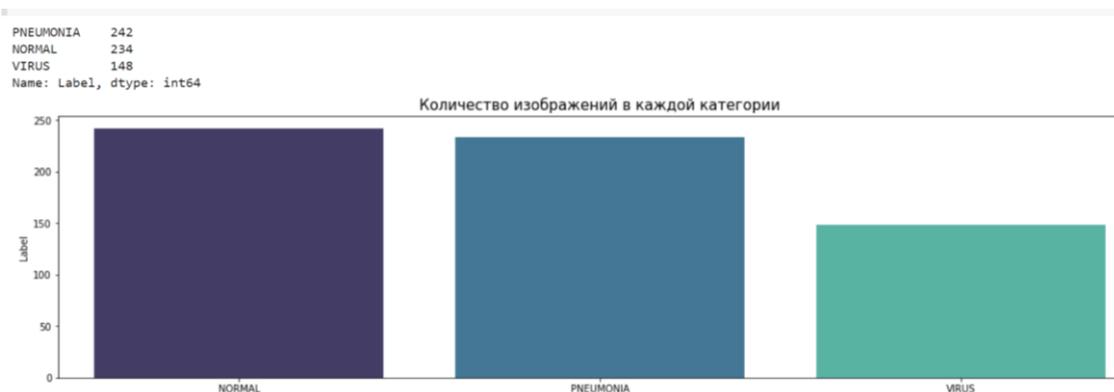


Рисунок 4.3 – Количество и соотношение изображений использованные для тестирования.

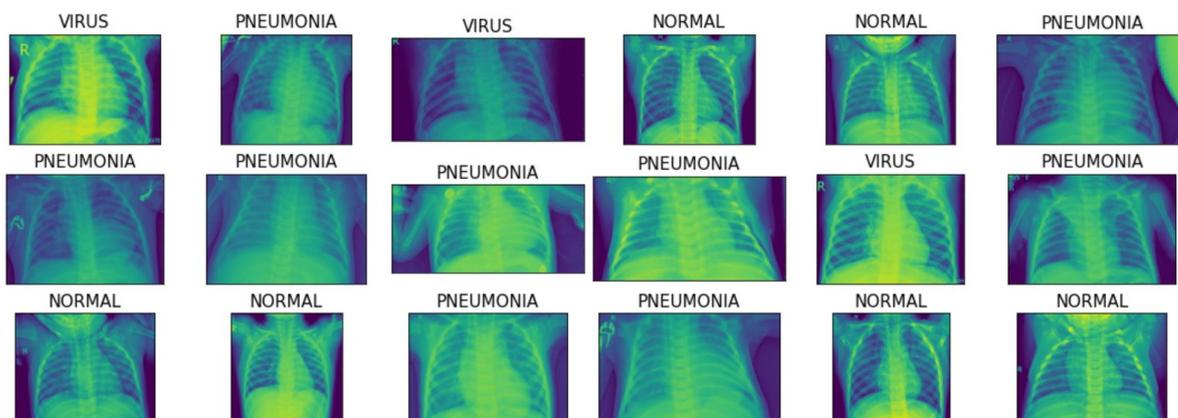


Рисунок 4.4 – Пример промаркированных рентгеновских снимков.

Для дальнейшего построения модели надо выбрать какие именно модели нам нужны. Так как главной целью этой работы является выбрать способ для автоматического определения диагноза по рентгеновскому снимку, то логично

будет выбрать несколько моделей, а также разные топологии внутри модели. Все выбранные модели и их топологии показаны на Рисунке 4.5

```
models = {
  "DenseNet121": {"model":tf.keras.applications.DenseNet121, "perf":0},
  "DenseNet169": {"model":tf.keras.applications.DenseNet169, "perf":0},
  "DenseNet201": {"model":tf.keras.applications.DenseNet201, "perf":0},
  "EfficientNetB0": {"model":tf.keras.applications.EfficientNetB0, "perf":0},
  "EfficientNetB1": {"model":tf.keras.applications.EfficientNetB1, "perf":0},
  "EfficientNetB2": {"model":tf.keras.applications.EfficientNetB2, "perf":0},
  "EfficientNetB3": {"model":tf.keras.applications.EfficientNetB3, "perf":0},
  "EfficientNetB4": {"model":tf.keras.applications.EfficientNetB4, "perf":0},
  "EfficientNetB5": {"model":tf.keras.applications.EfficientNetB4, "perf":0},
  "EfficientNetB6": {"model":tf.keras.applications.EfficientNetB4, "perf":0},
  "EfficientNetB7": {"model":tf.keras.applications.EfficientNetB4, "perf":0},
  "InceptionResNetV2": {"model":tf.keras.applications.InceptionResNetV2, "perf":0},
  "InceptionV3": {"model":tf.keras.applications.InceptionV3, "perf":0},
  "ResNet101": {"model":tf.keras.applications.ResNet101, "perf":0},
  "ResNet101V2": {"model":tf.keras.applications.ResNet101V2, "perf":0},
  "ResNet152": {"model":tf.keras.applications.ResNet152, "perf":0},
  "ResNet152V2": {"model":tf.keras.applications.ResNet152V2, "perf":0},
  "ResNet50": {"model":tf.keras.applications.ResNet50, "perf":0},
  "ResNet50V2": {"model":tf.keras.applications.ResNet50V2, "perf":0},
  "VGG16": {"model":tf.keras.applications.VGG16, "perf":0},
  "VGG19": {"model":tf.keras.applications.VGG19, "perf":0},
}
```

Рисунок 4.5 – Выбранные модели нейронной сети.

После выбора моделей последует их обучение. Полный код можно посмотреть в Приложении А, а время обучения всех моделей в Рисунке 4.6.

DenseNet121	trained in 81.14 sec
DenseNet169	trained in 87.38 sec
DenseNet201	trained in 92.69 sec
EfficientNetB0	trained in 75.33 sec
EfficientNetB1	trained in 80.51 sec
EfficientNetB2	trained in 84.18 sec
EfficientNetB3	trained in 87.61 sec
EfficientNetB4	trained in 94.28 sec
EfficientNetB5	trained in 96.42 sec
EfficientNetB6	trained in 94.42 sec
EfficientNetB7	trained in 94.28 sec
InceptionResNetV2	trained in 98.42 sec
InceptionV3	trained in 78.14 sec
ResNet101	trained in 91.18 sec
ResNet101V2	trained in 89.32 sec
ResNet152	trained in 104.63 sec
ResNet152V2	trained in 99.98 sec
ResNet50	trained in 78.78 sec
ResNet50V2	trained in 76.77 sec
VGG16	trained in 75.53 sec
VGG19	trained in 77.95 sec

Рисунок 4.6 –Время обучения выбранных моделей.

4.2 Результаты обучения

Оценка эффективности предложенного подхода выполняется с использованием 5 показателей оценки, рассмотренных в разделе 3.4.

Они приведены ниже, а таблице 4.1 и на рисунке 4.9.

	model	val_accuracy	accuracy (test set)	Training time (sec)
0	DenseNet169	0.8241	0.8093	70.04
1	ResNet50V2	0.8050	0.8013	61.62
2	InceptionV3	0.7629	0.7756	67.41
3	VGG16	0.7859	0.7628	72.41
4	ResNet101V2	0.8222	0.7532	70.07
5	ResNet152V2	0.8222	0.7516	75.13
6	VGG19	0.7228	0.7436	61.97
7	InceptionResNetV2	0.7954	0.7292	78.46
8	DenseNet121	0.8069	0.7099	375.47
9	DenseNet201	0.7648	0.6026	74.01
10	ResNet101	0.6635	0.5080	70.76
11	ResNet50	0.6195	0.4840	62.64
12	ResNet152	0.6616	0.4712	77.35
13	EfficientNetB6	0.4608	0.3878	76.43
14	EfficientNetB5	0.4608	0.3878	72.83
15	EfficientNetB4	0.4608	0.3878	75.26
16	EfficientNetB3	0.4608	0.3878	70.62
17	EfficientNetB2	0.4608	0.3878	68.46
18	EfficientNetB1	0.4608	0.3878	66.61
19	EfficientNetB0	0.4608	0.3878	64.11
20	EfficientNetB7	0.4608	0.3878	75.34

Таблица 4.1 – Результаты моделей после обучения.

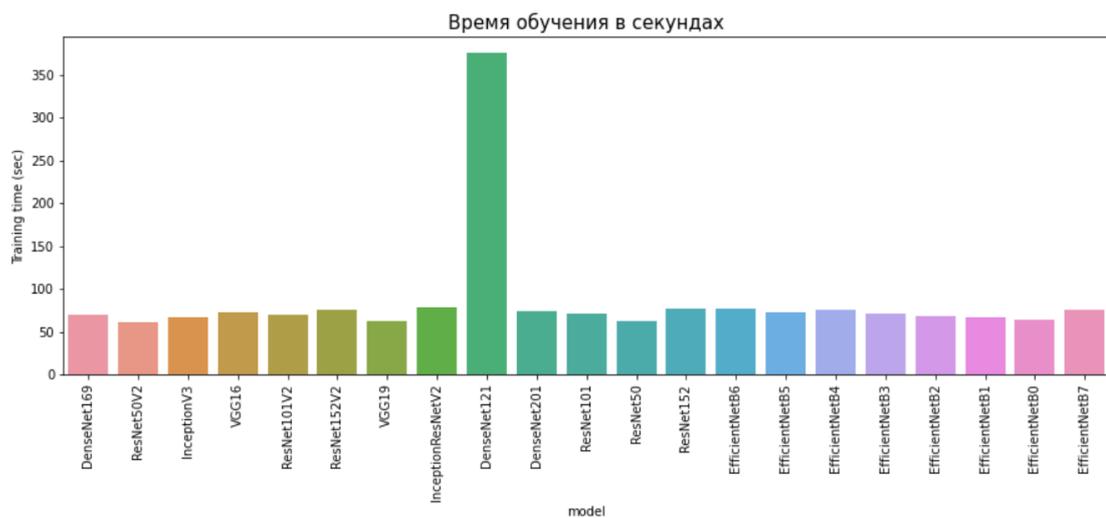


Рисунок 4.7 –Время обучения выбранных моделей.

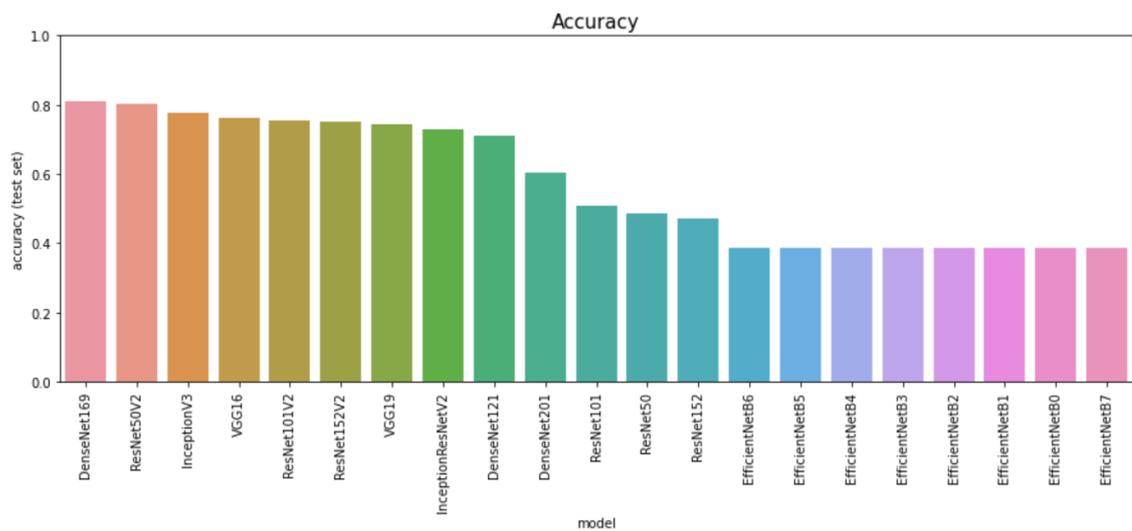


Рисунок 4.8 – Точность выбранных моделей.

Модель с лучшим показателем DenseNet-169, чью матрицу ошибок можно увидеть на Рисунке 4.9.

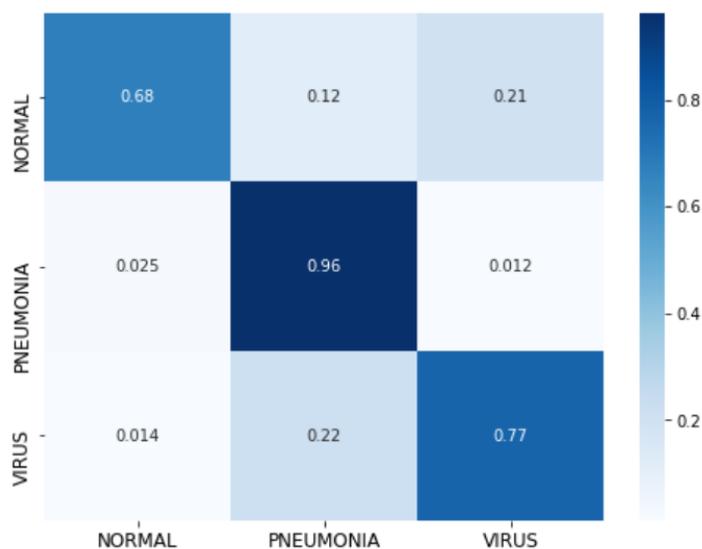


Рисунок 4.9 – Матрица ошибок модели DenseNet-169.

DenseNet-169: Нормальная точность с 82,41% классификацией путем ошибочной классификации 12% случаев как бактериальная пневмония и 21% случаев как вирусная пневмония. Для бактериальной пневмонии, 2,5% случаев классифицируются как нормальные, а 1,2% случаев вирусной пневмонии и вирусной пневмонии ошибочно классифицирует 1,4% случаев как нормальные и 22% случаев как бактериальная пневмония с общей средней точностью 80,93%.

В многоклассовых наборах данных ошибочно классифицированные экземпляры: нормальные как бактериальная и вирусная, бактериальная, как вирусная пневмония и вирусная пневмония ошибочно классифицируются как нормальные или бактериальные. Система иммунитета человека является основной причиной такой неправильной классификации. Потому что, когда люди поражаются вирусом или бактерией, они обычно не делают рентген грудной клетки каждый раз, и поражение с предыдущих заболеваний могут выявиться в следующий раз при рентгене.

Вирусная пневмония также ошибочно классифицируется как бактериальная пневмония из-за неясности рентгеновских снимков и высокой схожести с вирусной пневмонией при тяжёлых стадиях. В некоторых случаях предложенная система ошибочно классифицировала некоторые изображения COVID-19 как бактериальная пневмония, так же из-за сильной схожести на рентгеновских снимках.

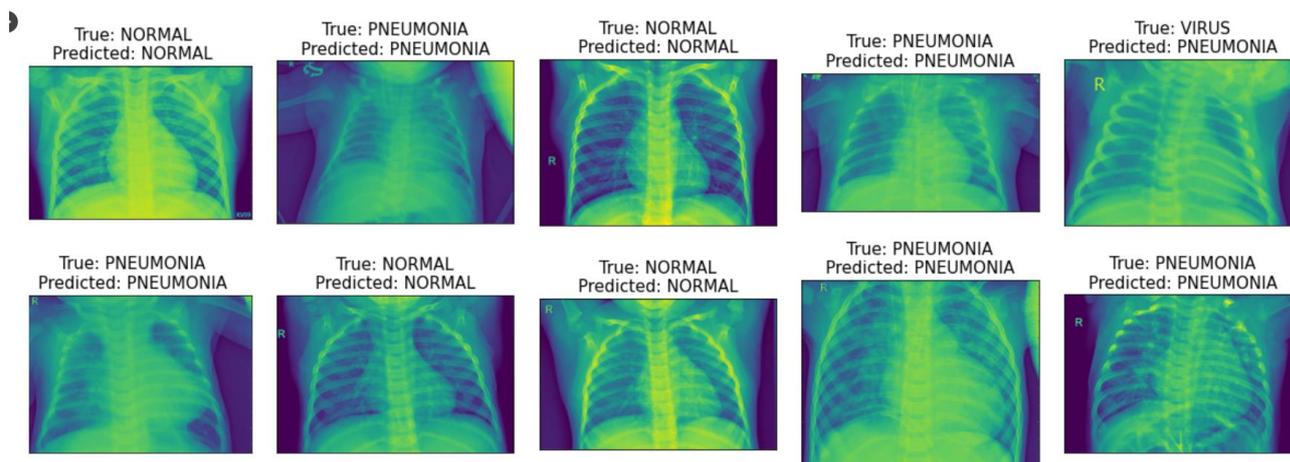


Рисунок 4.10 – Пример спрогнозированных диагнозов и сравнение с оригинальным диагнозом.

4.3 Использование предложенного метода

По результатам мы можем утверждать, что предлагаемый подход может быть использован как один из вспомогательных методов в области медицины для классификации рентгеновских снимков грудной клетки для выявления COVID-19. Машинное обучение можно использовать в качестве процесса предварительной оценки, чтобы снизить нагрузку на врача, выявить внутренние структуры, диагностировать и лечить заболевания на ранних стадиях. Если любое заболевание нижних дыхательных путей диагностируется на ранней стадии, его можно вылечить без сильных последствий или дальнейших поражений тканей.

Также можно применить этот подход для обработки изображений МРТ. Стандартный анализ МРТ требует нескольких часов компьютерного времени для анализа данных большого числа пациентов. Итак, применяя желаемую архитектуру CNN результат можно получить раньше. В области медицинской визуализации этот подход может ускорить процесс диагностики, чтобы ускорить процесс определения способа лечения. Таким образом, мы можем сделать вывод, что врачи и практикующие врачи могут использовать подход применения машинного обучения для диагностики, чтобы определить состояние органа и способы лечения, и что требуется для быстрого восстановления.

Заключение

Ранняя диагностика пациентов с COVID-19 важна для предотвращения распространения болезни на других. В данной работе, предоставляется подход глубокого обучения на основе CNN с использованием трансферного обучения, чтобы отличать пациентов с COVID-19(вирусная пневмония) от бактериальной пневмонией, и здоровых пациентов. Было развернуто двадцать предварительно обученных моделей CNN для изучения трансферного обучения и сделан вывод, что точная настройка предварительно обученного CNN модели могут успешно определить с точностью 82,41%.

Полученные высокоточные результаты могут быть полезны врачам и исследователям принимать решения в клинической практике.

В данной есть несколько ограничений, которые можно преодолеть в будущих исследованиях. В частности, более глубокий анализ требует гораздо больше данных о пациентах, особенно о тех, кто страдает COVID-19. Более интересный подход для будущих исследований будет сосредоточена на различении пациентов с легкими симптомами, а не с симптомами пневмонии, в то время как эти симптомы могут быть неточно визуализированы на рентгеновских снимках или могут вообще не визуализироваться.

Кроме того, эти методы можно использовать на более крупных наборах данных для решения других медицинских проблем, таких как рак, опухоли, и т.д., а также в других областях компьютерного зрения, таких как энергетика, сельское хозяйство и транспорт в ближайшее время.

Список использованной литературы

1. Никифоров В.В., Суранова Т.Г., Миронов А. Ю., Забозлаев Ф.Г. Новая коронавирусная инфекция (COVID-19): этиология, эпидемиология, клиника, диагностика, лечение и профилактика, Москва, 2020. - <https://epidbiomed-d.ru/professor/publikatsii/novaya-koronavirusnaya-infektsiya-covid-19-etiologya-epidemiologiya-klinika-diagnostika-lechenie-i/>
2. Всемирная организация здравоохранения: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19#:~:text=symptoms>
3. Rabab Yasin, Walaa Gouda. Chest X-ray findings monitoring COVID-19 disease course and severity. Egyptian Journal of Radiology and Nuclear Medicine, 2020 - <https://bmcpulmed.biomedcentral.com/articles/10.1186/s12890-020-01286-5>
4. Бактериальная пневмония: симптомы, лечение и профилактика - <https://ru.ncmhcs.org/bacterial-pneumonia-11788>
5. Joseph Paul Cohen, covid-chestxray-dataset <https://github.com/ieee8023/covid-chestxray-dataset>
6. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM (2017) ChestX-Ray8: hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, pp 3462–3471. <https://doi.org/10.1109/CVPR.2017.369>
7. Mooney P (2018) Chest X-ray images (Pneumonia). Kaggle Repository - <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>
8. R. J. Gillies, P. E. Kinahan, and H. Hricak, “Radiomics: Images are more than pictures, they are data,” Radiology, vol. 278, pp. 563–577, 2015.
9. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.
10. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional networks,” in Advances in Neural Processing Systems, vol. 25, 2012, pp. 1097–1105.
11. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
12. Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M. S. Lew, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27–48.

ПРИЛОЖЕНИЕ А

```
import numpy as np
import pandas as pd
from pathlib import Path
import os.path
import matplotlib.pyplot as plt
import tensorflow as tf
import seaborn as sns
from time import perf_counter
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
from IPython.display import Markdown, display

def printmd(string):
    display(Markdown(string))

from google.colab import drive
drive.mount('/content/drive', force_remount=True)

# Создаем список с путями к файлам для обучения и тестирования
dir_ = Path('/content/drive/MyDrive/chest_xray/train')
train_file_paths = list(dir_.glob(r'**/*.jpeg'))

dir_ = Path(r'/content/drive/MyDrive/chest_xray/test')
test_file_paths = list(dir_.glob(r'**/*.jpeg'))

dir_ = Path(r'/content/drive/MyDrive/chest_xray/val')
val_file_paths = list(dir_.glob(r'**/*.jpeg'))
# print(val_file_paths)
print('Операция успешно завершена')

def proc_img(filepath):
    """ Создаем датафрейм из изображений из нашего базы данных
    """

    labels = [str(filepath[i]).split("/")[-2] \
               for i in range(len(filepath))]

    filepath = pd.Series(filepath, name='Filepath').astype(str)
    labels = pd.Series(labels, name='Label')

    # Объединить пути к файлам и метки
    df = pd.concat([filepath, labels], axis=1)
```

Продолжение приложения А

```
# Перемешать DataFrame и сбросить индекс
df = df.sample(frac=1, random_state=0).reset_index(drop = True)

return df

train_df = proc_img(train_file_paths)
val_df = proc_img(val_file_paths)

train_df = pd.concat([train_df, val_df]).reset_index(drop = True)
test_df = proc_img(test_file_paths)

print(f'Количество изображений для обучения модели: {train_df.shape[0]}')
print(f'Количество изображений для тестирования модели: {test_df.shape[0]}')
print(f'Количество изображений для проверки: {val_df.shape[0]}\n')

print(f'Количество видов диагнозов: {len(train_df.Label.unique())}\n')
print(f'Диагнозы: {train_df.Label.unique()}')

# Датафрейм
train_df.head(5)

# Отображение количества изображений каждой категории в обучающем наборе
vc = train_df['Label'].value_counts()
print(vc)

plt.figure(figsize=(20,5))
sns.barplot(x = sorted(vc.index), y = vc, palette = "mako")
plt.title("Количество изображений в каждой категории", fontsize = 15)
plt.show()

# Отображение количества изображений каждой категории в обучающем наборе
vc = test_df['Label'].value_counts()
print(vc)

plt.figure(figsize=(20,5))
sns.barplot(x = sorted(vc.index), y = vc, palette = "mako")
plt.title("Количество изображений в каждой категории", fontsize = 15)
plt.show()

# Показать несколько изображений набора данных
fig, axes = plt.subplots(nrows=4, ncols=6, figsize=(15, 7),
                          subplot_kw={'xticks': [], 'yticks': []})
```

Продолжение приложения А

```
for i, ax in enumerate(axes.flat):
    ax.imshow(plt.imread(train_df.Filepath[i]))
    ax.set_title(train_df.Label[i], fontsize = 15)
plt.tight_layout(pad=0.5)
plt.show()

def create_gen():

    train_generator = tf.keras.preprocessing.image.ImageDataGenerator(
        preprocessing_function=tf.keras.applications.mobilenet_v2.preproce
ss_input,
        validation_split=0.1
    )

    test_generator = tf.keras.preprocessing.image.ImageDataGenerator(
        preprocessing_function=tf.keras.applications.mobilenet_v2.preproce
ss_input
    )

    train_images = train_generator.flow_from_dataframe(
        dataframe=train_df,
        x_col='Filepath',
        y_col='Label',
        target_size=(224, 224),
        color_mode='rgb',
        class_mode='categorical',
        batch_size=32,
        shuffle=True,
        seed=0,
        subset='training',
    )

    val_images = train_generator.flow_from_dataframe(
        dataframe=train_df,
        x_col='Filepath',
        y_col='Label',
        target_size=(224, 224),
        color_mode='rgb',
        class_mode='categorical',
        batch_size=32,
        shuffle=True,
        seed=0,
        subset='validation',
    )
```

Продолжение приложения А

```
test_images = test_generator.flow_from_dataframe(
    dataframe=test_df,
    x_col='Filepath',
    y_col='Label',
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=32,
    shuffle=False
)

return train_generator, test_generator, train_images, val_images, test_images

def get_model(model):
    # Загрузите готовую модель
    kwargs = {
        'input_shape': (224, 224, 3),
        'include_top': False,
        'weights': 'imagenet',
        'pooling': 'avg'
    }

    pretrained_model = model(**kwargs)
    pretrained_model.trainable = False

    inputs = pretrained_model.input

    x = tf.keras.layers.Dense(128, activation='relu')(pretrained_model.output)
    x = tf.keras.layers.Dense(128, activation='relu')(x)

    outputs = tf.keras.layers.Dense(3, activation='softmax')(x)

    model = tf.keras.Model(inputs=inputs, outputs=outputs)

    model.compile(
        optimizer='adam',
        loss='categorical_crossentropy',
        metrics=['accuracy']
    )

    return model

# Dictionary with the models
models = {
    "DenseNet121": {"model": tf.keras.applications.DenseNet121, "perf": 0},
    "DenseNet169": {"model": tf.keras.applications.DenseNet169, "perf": 0},
    "DenseNet201": {"model": tf.keras.applications.DenseNet201, "perf": 0},
    "EfficientNetB0": {"model": tf.keras.applications.EfficientNetB0, "perf": 0},
    "EfficientNetB1": {"model": tf.keras.applications.EfficientNetB1, "perf": 0},
}
```

Продолжение приложения А

```
"EfficientNetB2": {"model":tf.keras.applications.EfficientNetB2, "perf":0
},
  "EfficientNetB3": {"model":tf.keras.applications.EfficientNetB3, "perf
":0},
  "EfficientNetB4": {"model":tf.keras.applications.EfficientNetB4, "perf
":0},
  "EfficientNetB5": {"model":tf.keras.applications.EfficientNetB4, "perf
":0},
  "EfficientNetB6": {"model":tf.keras.applications.EfficientNetB4, "perf
":0},
  "EfficientNetB7": {"model":tf.keras.applications.EfficientNetB4, "perf
":0},
  "InceptionResNetV2": {"model":tf.keras.applications.InceptionResNetV2,
"perf":0},
  "InceptionV3": {"model":tf.keras.applications.InceptionV3, "perf":0},
  "ResNet101": {"model":tf.keras.applications.ResNet101, "perf":0},
  "ResNet101V2": {"model":tf.keras.applications.ResNet101V2, "perf":0},
  "ResNet152": {"model":tf.keras.applications.ResNet152, "perf":0},
  "ResNet152V2": {"model":tf.keras.applications.ResNet152V2, "perf":0},
  "ResNet50": {"model":tf.keras.applications.ResNet50, "perf":0},
  "ResNet50V2": {"model":tf.keras.applications.ResNet50V2, "perf":0},
  "VGG16": {"model":tf.keras.applications.VGG16, "perf":0},
  "VGG19": {"model":tf.keras.applications.VGG19, "perf":0},

}

train_generator,test_generator,train_images,val_images,test_images=create_
gen()
print('\n')

for name, model in models.items():
    m = get_model(model['model'])
    models[name]['model'] = m

    start = perf_counter()

    history = m.fit(train_images,validation_data=val_images,epochs=1,verbo
se=0)

    duration = perf_counter() - start
    duration = round(duration,2)
    models[name]['perf'] = duration
    print(f"{name:20} trained in {duration} sec")
```

Продолжение приложения А

```
val_acc = history.history['val_accuracy']
models[name]['val_acc'] = [round(v,4) for v in val_acc]

for name, model in models.items():

    pred = models[name]['model'].predict(test_images)
    pred = np.argmax(pred,axis=1)

    labels = (train_images.class_indices)
    labels = dict((v,k) for k,v in labels.items())
    pred = [labels[k] for k in pred]

    y_test = list(test_df.Label)
    acc = accuracy_score(y_test,pred)
    models[name]['acc'] = round(acc,4)
# Создать датафрэйм с результатами
models_result = []

for name, v in models.items():
    models_result.append([ name, models[name]['val_acc'][-1],
                          models[name]['acc'],
                          models[name]['perf']])

df_results = pd.DataFrame(models_result,
                           columns = ['model','val_accuracy','accuracy (test set)','Training time (sec)'])
df_results.sort_values(by='accuracy (test set)', ascending=False, inplace=True)
df_results.reset_index(inplace=True,drop=True)
df_results
plt.figure(figsize = (15,5))
sns.barplot(x = 'model', y = 'accuracy (test set)', data = df_results)
plt.title('Accuracy', fontsize = 15)
plt.ylim(0,1)
plt.xticks(rotation=90)
plt.show()

plt.figure(figsize = (15,5))
sns.barplot(x = 'model', y = 'Training time (sec)', data = df_results)
plt.title('Время обучения в секундах', fontsize = 15)
plt.xticks(rotation=90)
plt.show()
```

Продолжение приложения А

```
acc = df_results.iloc[0]['accuracy (test set)']
best_model = df_results.iloc[0]['model']
printmd(f'# Лучшая модель: {best_model}')
printmd(f'# Точность лучшей модели: {acc * 100:.2f}%')

pred = models[best_model]['model'].predict(test_images)
pred = np.argmax(pred,axis=1)

labels = (train_images.class_indices)
labels = dict((v,k) for k,v in labels.items())
pred = [labels[k] for k in pred]

y_test = list(test_df.Label)
acc = accuracy_score(y_test,pred)
models[name]['acc'] = round(acc,4)

# Confusion matrix для лучшей модели
from sklearn.metrics import confusion_matrix
cf_matrix = confusion_matrix(y_test, pred, normalize='true')
plt.figure(figsize = (8,6))
sns.heatmap(cf_matrix, annot=True, cmap= 'Blues', xticklabels = sorted(set
(y_test)), yticklabels = sorted(set(y_test)))
plt.title('Normalized Confusion Matrix\n', fontsize = 23)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()

from sklearn.metrics import classification_report
print(classification_report(y_test,pred))

fig, axes = plt.subplots(nrows=4, ncols=6, figsize=(20, 12),
                        subplot_kw={'xticks': [], 'yticks': []})

for i, ax in enumerate(axes.flat):
    ax.imshow(plt.imread(test_df.Filepath.iloc[i]))
    ax.set_title(f"Диагноз: {test_df.Label.iloc[i].split('_')[0]}\nСпрогно
зированный диагноз: {pred[i].split('_')[0]}", fontsize = 15)
plt.tight_layout()
plt.show()
```

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Отзыв научного руководителя

Дипломная работа

Арманкызы Рената

5B070300 – Информационные системы

Тема: Поиск аномалий в рентгеновских снимках при помощи глубокого машинного обучения

Дипломная работа на тему «Поиск аномалий в рентгеновских снимках при помощи глубокого машинного обучения» – представляет собой выпускную квалификационную работу по специальности 5B070300 – «Информационные системы». Пояснительная записка состоит из введения, четырех глав, заключения, списка использованных источников и приложения.

Студент полностью выполнил поставленные задачи дипломной работы и показал владение современными технологиями в предметной области. В процессе подготовки и написания работы студентка соблюдала сроки выполнения календарного графика и проявила отличные навыки в работе с теоретическими источниками и получила результаты использования 20 разных моделей глубокого обучения для классификации заболеваний нижних дыхательных путей.

Дипломная работа на тему «Поиск аномалий в рентгеновских снимках при помощи глубокого машинного обучения» выполнена на достаточном профессиональном уровне и содержит все необходимые сведения и материалы для такого рода работ. К замечаниям следует отнести незначительные стилистические ошибки.

Практическая значимость работы заключается в выборе наилучшей модели для быстрого обнаружения изменений в легких при возникновении заболеваний или при отслеживании процесса протекания болезни. В третьей главе описана общее описание используемых моделей, базы данных, экспериментальной установки и показателей производительности, по которым будут оцениваться модели. В четвертой части представлены результаты и оценка моделей.

Считаю, что дипломная работа на тему «Поиск аномалий в рентгеновских снимках при помощи глубокого машинного обучения» соответствует требованиям, предъявляемым к выпускным квалификационным работам по специальности 5B070300 – «Информационные системы». Автор работы Арманкызы Рената заслуживает присвоения академической степени бакалавра.

Научный руководитель

Магистр технических наук., Лектор

«29» мая 2021



Аристомбаева М.Т.

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Арманкызы Рената

Название: Поиск аномалий в рентгеновских снимках при помощи глубокого машинного обучения

Координатор: Аристомбаева М.Т

Коэффициент подобия 1: 2,12

Коэффициент подобия 2: 1,19

Замена букв:0

Интервалы:0

Микропробелы:2

Белые знаки:0

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;

обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы, по существу, и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;

обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование: После анализа отчёта по плагиату и работы дипломника выявлено, что заимствования являются добросовестными и не обладают признаками плагиата, поскольку связаны с применением общеизвестных терминов и также названий общедоступных программ и файлов.

Дата 28.05.2021



Сейлова Н.А., подпись зав.кафедрой

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Арманкызы Рената

Название: Поиск аномалий в рентгеновских снимках при помощи глубокого машинного обучения

Координатор: Аристомбаева М.Т

Коэффициент подобия 1: 2,12

Коэффициент подобия 2: 1,19

Замена букв:0

Интервалы:0

Микропробелы:2

Белые знаки:0

После анализа Отчета подобия констатирую следующее:

обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;

обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;

обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование: Заимствования являются добросовестными и не обладают признаками плагиата, объясняются использованием общепринятой терминологией, а также названиями программ и файлов.

Дата 27.05.2021



Аристомбаева М.Т.,

Подпись Научного руководителя