Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

Классификация изображений легких

Студент ИУ5-61Б (Группа) _____ (Подпись, дата) Шимолина П.К. (И.О.Фамилия)

Руководитель _____ (Подпись, дата) Гапанюк Ю.Е. (И.О.Фамилия)

Москва, 2023 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой ИУ5
(Индекс)

(И.О.Фамилия)

« ____ » _____ 20 ____ г.

З А Д А Н И Е
на выполнение научно-исследовательской работы

по теме Классификация изображений легких

Студент группы ИУ5-61Б

Шимолина Полина Кирилловна
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)
Исследовательская

Источник тематики (кафедра, предприятие, НИР)
НИР

График выполнения НИР: 25% к ___ нед., 50% к ___ нед., 75% к __ нед., 100% к ___ нед.

Техническое задание Исследовать методы машинного обучения для решения задачи классификации изображений

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на 27 листах формата А4.
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания «13» февраля 2023 г.

Руководитель НИР _____ Гапанюк Ю.Е.
(Подпись, дата) (И.О.Фамилия)

Студент _____ Шимолина П.К.
(Подпись, дата) (И.О.Фамилия)

Оглавление

Введение

Пневмония — это острое или хроническое воспаление легких, которое может быть вызвано различными факторами, включая бактерии, вирусы, грибы и другие возбудители. Диагностика пневмонии является одной из наиболее важных задач в медицине, так как раннее выявление заболевания и назначение эффективной терапии способны спасти жизнь пациента.

Целью настоящего исследования является разработка и обучение максимально точной модели машинного обучения для классификации легких на два класса: больные пневмонией и здоровые.

Задачи и этапы работы:

Для достижения поставленной цели были определены следующие задачи:

1. Собрать и подготовить набор данных изображений легких, включающий в себя изображения здоровых и больных пневмонией.
2. Разделить набор данных на тренировочный и тестовый наборы.
3. Подготовить изображения для обучения модели, в том числе привести их к одному размеру и нормализовать пиксельные значения.
4. Обучить несколько моделей на тренировочном наборе данных.
5. Протестировать модели на тестовом наборе данных
6. Выбрать метрики для оценки модели
7. Оценить качество обученных моделей с помощью метрик.

Актуальность работы:

В современном мире существует большое количество методов и алгоритмов для диагностики пневмонии. Однако, с развитием машинного обучения и глубоких нейронных сетей, возможности автоматической диагностики значительно увеличились. Например, в Москве уже начали практику выявления рассеянного склероза на магнитно-резонансной томографии головного мозга с помощью нейросетей. Разработка и обучение модели машинного обучения для классификации легких на два класса: больные пневмонией и здоровые, может значительно повысить точность и

скорость диагностики, что имеет большое значение для здоровья и жизни пациентов.

1. Постановка задачи

В настоящее время существует большое количество методов и технологий, используемых для анализа медицинских изображений, включая изображения легких пациентов с пневмонией. Однако, ручная интерпретация и анализ таких изображений может быть трудоемкой и неточной, что делает необходимым разработку автоматизированных методов для диагностики пневмонии на основе машинного обучения.

Цель настоящей работы - разработка и обучение модели машинного обучения для классификации легких на два класса: больные пневмонией и здоровые.

Актуальность решаемой задачи подтверждается множеством исследований, проводимых в области медицинской диагностики. Одним из ярких примеров таких исследований является работа [1], в которой авторы использовали глубокие нейронные сети для диагностики пневмонии на основе медицинских изображений. В ходе исследования была разработана модель, позволяющая диагностировать пневмонию с точностью почти 90%.

Кроме того, в работе [2] была проведена оценка эффективности различных методов диагностики пневмонии на основе медицинских изображений. В ходе исследования было выявлено, что использование архитектуры InceptionV3 позволяет достичь более высокой точности диагностики по сравнению с другими архитектурами.

Среди других исследований, связанных с диагностикой пневмонии на основе медицинских изображений, можно упомянуть работу [3].

В каждой из этих работ были использованы различные методы машинного обучения, включая глубокие нейронные сети, для классификации легких на два класса: больные пневмонией и здоровые. В ходе исследований были достигнуты высокие показатели точности диагностики.

Анализ аналогичных решений:

В статье "A CNN to Classify Pneumonia—Step by Step using PyTorch" [1] описывается использование сверточных нейронных сетей для классификации пневмонии на основе датасета Chest X-Ray.

Автор статьи использовал библиотеку PyTorch для обучения своей модели, которая состоит из двух сверточных слоев и двух полносвязных слоев. Предобработка изображений включала в себя изменение размера изображений до 224x224 пикселей и нормализацию пикселей к диапазону от 0 до 1.

Для оценки результатов использовались метрики точности (accuracy), чувствительности (recall), специфичности (specificity) и F1-мера (F1-score). Результаты экспериментов показали, что сверточная нейронная сеть дает хорошие результаты в классификации пневмонии на основе датасета Chest X-Ray, достигая точности в районе 88-90% в зависимости от выбранной метрики.

В статье "Detecting Pneumonia in X-ray Images Using Convolutional Neural Networks" [2] авторы используют сверточные нейронные сети для диагностики пневмонии по рентгеновским снимкам. Используется набор данных Chest X-Ray8, содержащий 108,948 изображений, взятых у 32,717 пациентов, с метками "нормальный" и "пневмония".

Предобработка данных включает изменение размеров изображений до 256x256 и нормализацию пикселей в диапазоне [0,1]. Разведочный анализ включает визуализацию нескольких примеров изображений каждого класса, а также построение графиков распределения классов в обучающем и тестовом наборах.

Авторы использовали три различных архитектуры сверточных нейронных сетей (VGG16, ResNet50 и InceptionV3) и сравнили их результаты на тестовом наборе данных. В качестве метрик использовались точность (accuracy), чувствительность (recall), специфичность (specificity) и площадь под ROC-кривой (ROC AUC). Наиболее высокие результаты были достигнуты с помощью архитектуры InceptionV3, с точностью 0.85,

чувствительностью 0.93, специфичностью 0.80 и площадью под ROC-кривой 0.93.

В статье "Pneumonia Diagnosis using CNNs" [3] описывается использование сверточных нейронных сетей для классификации рентгеновских изображений грудной клетки на наличие пневмонии.

Методы, используемые в статье, включают в себя применение сверточных нейронных сетей (CNN), аугментации данных и оптимизации параметров модели.

Для обучения и тестирования модели используется набор данных Chest X-Ray Images (Pneumonia), содержащий рентгеновские изображения грудной клетки с диагнозами нормального состояния, бактериальной пневмонии и вирусной пневмонии.

Метрики, используемые для оценки производительности модели, включают в себя точность (accuracy), чувствительность (sensitivity), специфичность (specificity) и F1-меру (F1-score).

В предобработке данных используется нормализация значений пикселей, а также применение аугментации данных, включающей в себя случайное поворот, изменение масштаба и отражение.

В разведочном анализе данных была проведена визуализация рентгеновских изображений для каждой из категорий диагнозов, а также построены матрицы корреляции между признаками. Также были проанализированы распределения классов в обучающей и тестовой выборках.

Во всех рассмотренных работах были достигнуты высокие показатели точности диагностики пневмонии на основе медицинских изображений.

Актуальность решаемой задачи заключается в том, что пневмония является распространенным и опасным заболеванием легких, которое может привести к тяжелым осложнениям и смерти. Быстрая и точная диагностика пневмонии является критически важной для эффективного лечения и предотвращения развития осложнений. Также, в свете пандемии COVID-19,

разработка более точных и эффективных методов диагностики является особенно актуальной.

Анализ аналогичных решений показал, что для решения данной задачи широко применяются глубокие нейронные сети, такие как VGG, ResNet, Inception и другие. Наборы данных, используемые для обучения моделей, включают в себя медицинские изображения легких, полученные с помощью различных методов, включая рентгеновскую томографию и компьютерную томографию. Метрики, используемые для оценки качества обученных моделей, включают в себя точность (accuracy), чувствительность (recall), специфичность (specificity) и F1-мера (F1-score).

В сравнении с рассмотренными работами, данная работа имеет некоторые отличия. В частности, мы будем использовать набор данных, состоящий из медицинских изображений легких, полученных с помощью рентгеновской томографии, и обучать несколько моделей с целью сравнить эффективность. Оценка качества модели будет проводиться с помощью метрики точности (precision), полноты (recall) и F-меры (f1-score). Ожидается, что разработанная модель позволит достичь высокой точности.

2. Используемый метод

В работе используется метод сверточной нейронной сети (Convolutional Neural Network, CNN) для классификации изображений легких на 2 класса: больные пневмонией и здоровые.

CNN — это одна из наиболее распространенных архитектур нейронных сетей для работы с изображениями. Она состоит из нескольких слоев, каждый из которых выполняет определенную функцию.

Слой свертки (Conv2D) извлекает признаки из входных изображений. Conv2D проходит по изображению, используя фильтры (ядро свертки), которые совершают локальные операции свертки с небольшими фрагментами входного изображения, чтобы вычислить значения для нового признакового слоя. Затем сверточный слой может использовать несколько таких фильтров, чтобы извлечь различные признаки из входного изображения.

Слой объединения (MaxPooling2D) выполняет операцию подвыборки (pooling) с целью уменьшения размера изображения и извлечения наиболее значимых признаков. Он выбирает максимальное значение в определенной области входного изображения и заменяет эту область на это значение. Например, если установлено значение параметра pool_size = a, то каждый блок аха пикселей во входном изображении заменяется на один пиксель с максимальным значением из этого блока.

Слой BatchNormalisation используется для нормализации активаций между слоями и ускорения процесса обучения. Кроме того, BatchNormalization уменьшает переобучение, ускоряет сходимость обучения и может повысить общую производительность нейронной сети. BatchNormalization включает несколько этапов: сначала, он центрирует и нормализует каждый признак внутри батча, затем масштабирует каждый признак с помощью двух настраиваемых параметров (gamma и beta), которые позволяют нейронной сети узнавать, какие функции лучше использовать и какие нет. Наконец, BatchNormalization применяет линейное преобразование,

чтобы получить нормализованные активации, которые передаются в следующий слой.

Слой Flatten преобразует многомерный массив в одномерный, сохраняя при этом все значения. Например, если входной массив имеет размерность (batch_size, height, width, channels), то Flatten преобразует его в одномерный массив размерности (batch_size, height * width * channels). Это необходимо для того, чтобы передать данные на вход полносвязному слою, который ожидает одномерный вектор.

Слой Dense - это полносвязный слой, который используется для выполнения линейной операции над входными данными и создания выходного вектора с фиксированным размером. Слой Dense с функцией активации sigmoid используется для решения задач бинарной классификации. Функция активации sigmoid принимает на вход значение в диапазоне от минус бесконечности до плюс бесконечности и преобразует его в значение в диапазоне от 0 до 1. Это значение можно интерпретировать как вероятность того, что входной образ принадлежит к положительному классу. Если вероятность больше или равна пороговому значению (обычно 0,5), то образ относится к положительному классу, в противном случае - к отрицательному.

Слой Dropout помогает бороться с переобучением модели путем случайного обнуления (выключения) некоторых нейронов во время обучения. Каждый нейрон в слое Dropout имеет вероятность оставаться активным или быть обнуленным, которая задается в качестве параметра dropout rate. Во время обучения, для каждого входного примера, каждый нейрон может быть обнулен с заданной вероятностью

Для обучения моделей был использован датасет, состоящий из изображений легких пациентов, разделенных на 2 категории: тренировочный и тестовый. Входные изображения были приведены к единому размеру и подвергнуты нормализации.

После обучения модель была протестирована на тестовом наборе данных, и ее точность была оценена.

В качестве метрик выберем Recall, Precision, F1-score, так как в данных присутствует небольшой, но приемлемый дисбаланс классов, и нам важно посмотреть, как точно модель предсказывает само наличие пневмонии.

3. Выполнение работы

В начале загружаем датасет с изображениями из отдельных директорий, используя функции библиотеки OpenCV.

Образцы изображений:

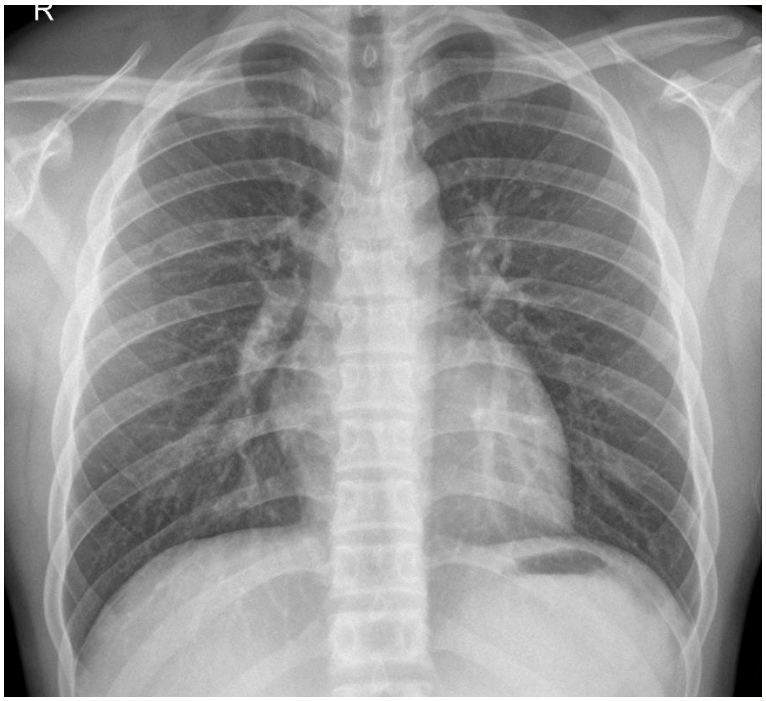

Рисунок 1 - Образец изображения нормальных легких

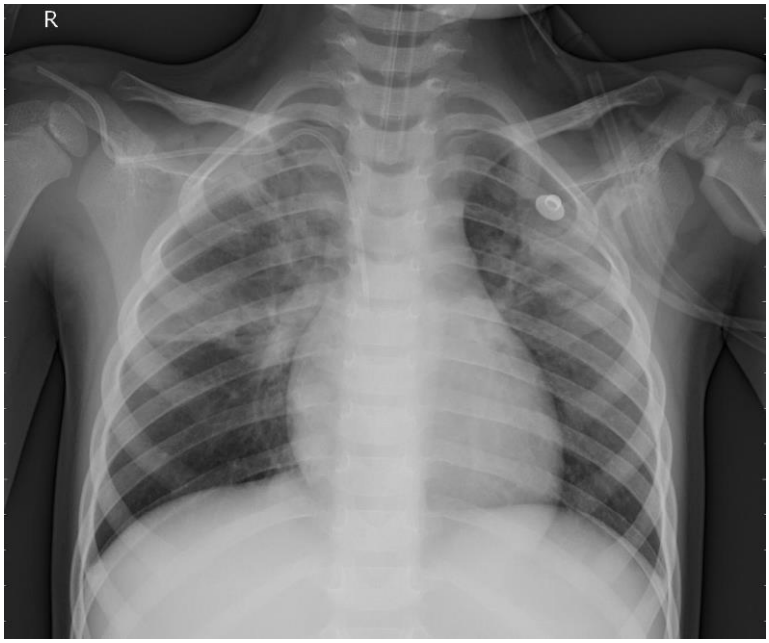

Рисунок 2 - Образец изображения легких больных пневмонией

Смотрим соотношение признаков:

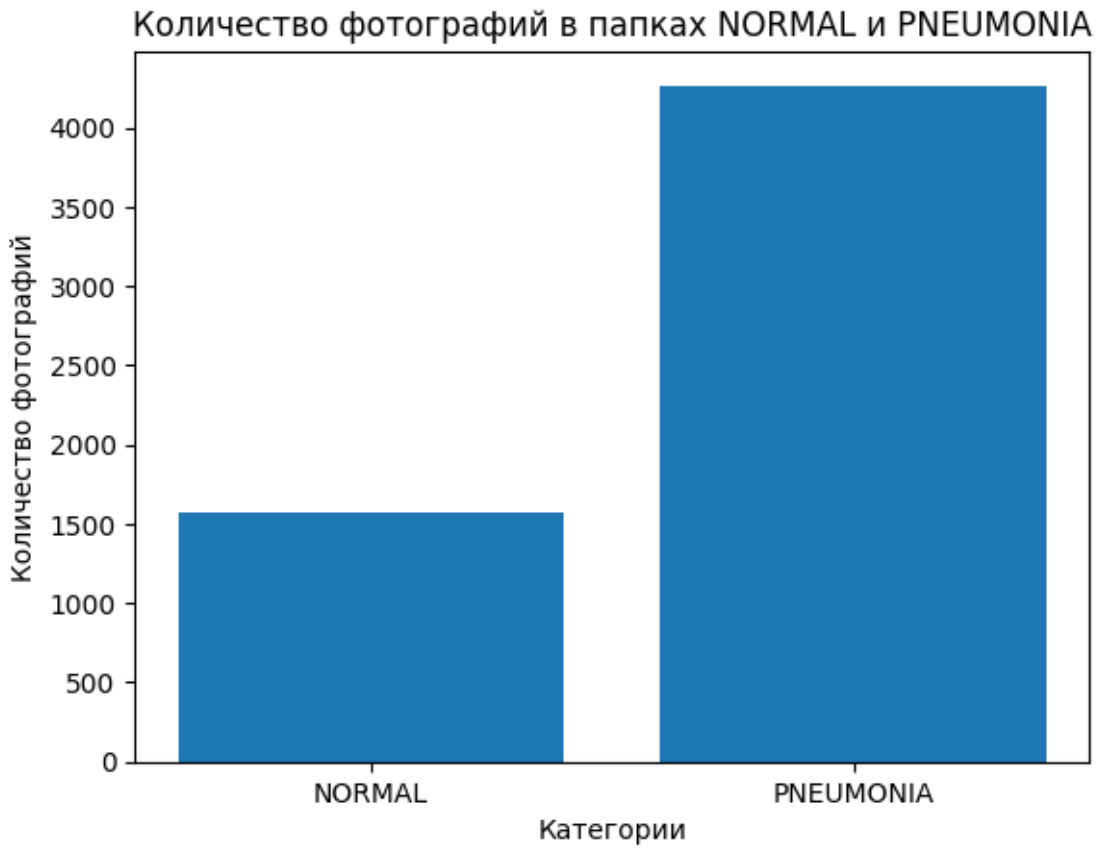

Рисунок 3 - Гистограмма соотношения классов

В результате получаем массив, содержащий изображения в градации серого и их классы. Разделяем массив на два: X - изображение и Y - класс. Преобразовываем в numpy массив и устанавливаем размер изображений. Выполняем нормализацию пиксели в диапазоне от 0 до 1.

Далее с помощью класса Sequential() определяем модель как последовательность слоев. Такая модель нейронной сети подразумевает, что каждый выходной слой является входом для следующего слоя, т.е. информация "протекает" через каждый слой в последовательном порядке.

Модель включает в себя следующие слои:

1. Сверточный слой (Conv2D) с 32 фильтрами размером (3,3) и функцией активации ReLU, принимающий входное изображение размером IMG_SIZE x IMG_SIZE x 1 (одноканальное изображение).
2. Слой подвыборки (MaxPooling2D) с размером окна (2,2), который уменьшает размерность выхода сверточного слоя вдвое.

3. Аналогичный сверточный слой с 64 фильтрами и функцией активации ReLU.
4. Аналогичный слой подвыборки.
5. Еще один сверточный слой с 128 фильтрами и функцией активации ReLU.
6. Еще один слой подвыборки.
7. Слой "сплющивания" (Flatten), который преобразует выходы сверточных слоев в одномерный вектор.
8. Полносвязный слой (Dense) с 64 нейронами и функцией активации ReLU.
9. Выходной слой с одним нейроном и функцией активации sigmoid, который выдает вероятность принадлежности изображения к одному из двух классов.

Затем создаем свои метрики, т.к. метрики accuracy недостаточно для определения точности модели, в которой присутствует дисбаланс классов. Создаем recall, precision и f1-score, регистрируем их и используем при компиляции в качестве параметра metrics.

Метод compile принимает следующие параметры:

- Optimizer – оптимизатор, который будет использоваться для обновления весов в процессе обучения. Используем оптимизатор Adam с коэффициентом обучения равным 0.001. Оптимизатор Adam - это адаптивный оптимизатор, который показывает хорошие результаты для многих типов моделей нейронных сетей.
- Loss – функция потерь, которую использует модель для измерения ошибки в процессе обучения. Используем binary_crossentropy, которая подходит для бинарной классификации, где модель выдает вероятность принадлежности к одному из двух классов.
- Metrics – список метрик, которые будут использоваться для оценки качества работы модели в процессе обучения и тестирования.

В результате получается модель нейронной сети, которая принимает на вход нормализованные изображения и выдает на выходе вероятность принадлежности изображения к классу.

Определяем размер пакетов BATCH_SIZE = 32, количество эпох EPOCHS = 50.

Определяем train_datagen, как объект класса ImageDataGenerator, train_generator – генератор потока данных, который используется для обучения модели с помощью метода fit в Keras. Он создает пакеты данных batches из обучающих изображений X_train и соответствующих меток классов y_train, которые будут использоваться для обновления весов модели в каждой эпохе обучения. Размер каждого пакета определяется параметром batch_size.

Задаем критерий остановки обучения: прекращаем обучение когда функция потерь не улучшается в течение 5 эпох, восстанавливаем веса модели на лучшей эпохе, т.е. с наименьшей функцией потерь.

Информация о модели:

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)           (None, 148, 148, 32)      320

 max_pooling2d_3 (MaxPooling  (None, 74, 74, 32)       0
 2D)

 conv2d_4 (Conv2D)           (None, 72, 72, 64)        18496

 max_pooling2d_4 (MaxPooling  (None, 36, 36, 64)       0
 2D)

 conv2d_5 (Conv2D)           (None, 34, 34, 128)       73856

 max_pooling2d_5 (MaxPooling  (None, 17, 17, 128)      0
 2D)

 flatten_1 (Flatten)         (None, 36992)             0

 dense_2 (Dense)             (None, 64)                2367552

 dense_3 (Dense)             (None, 1)                 65

=================================================================
Total params: 2,460,289
Trainable params: 2,460,289
Non-trainable params: 0
_________________________________________________________________
```

Рисунок 4 - Информация о модели 1

Обучаем модель. Модель остановилась на 11 эпохах. Лучшая эпоха – 6.

Оцениваем качество лучшей модели на тестовых данных:

```
Test loss: 0.1271
Test Accuracy: 0.9555
Test F1-score: 0.9686
Test recall: 0.9762
Test Precision: 0.9622
```

Построим график зависимости значения метрик точности от эпохи обучения для обучающего и проверочного наборов данных. Код строит график с двумя кривыми: "Training" (кривая обучения) и "Validation" (кривая проверки).

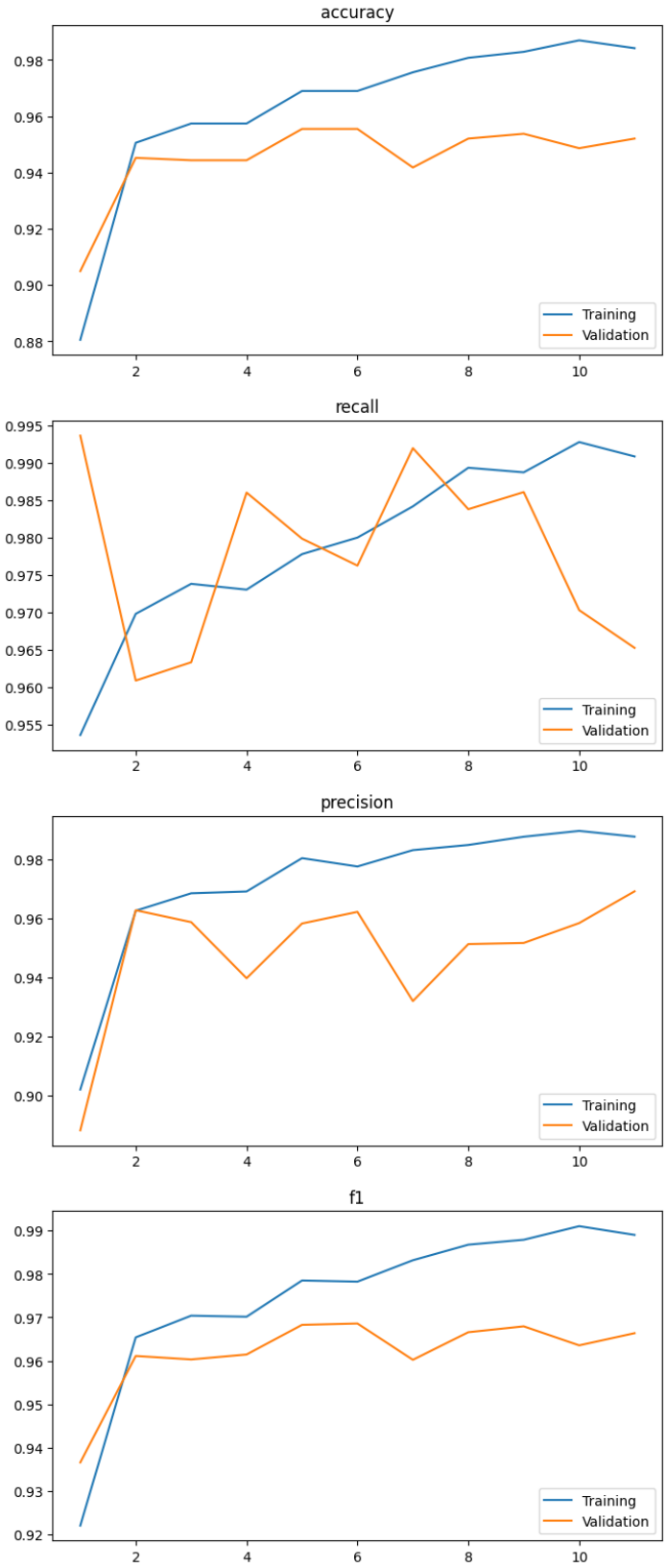

Рисунок 5 - Графики изменения метрик в зависимости от эпохи для модели 1

Попробуем увеличить точность модели. Добавим слои BatchNormalisation. Также зададим параметры аугментации:

- rotation_range = 20 - диапазон в градусах для поворота изображения. То есть изображения могут поворачиваться на случайный угол в диапазоне от -20 до 20 градусов.
- zoom_range = 0.15 - диапазон для случайного масштабирования изображения. Изображения могут увеличиваться или уменьшаться на случайный коэффициент в диапазоне от 0,85 до 1,15.
- width_shift_range = 0.2 и height_shift_range = 0.2 - диапазон для случайного смещения изображения по горизонтали и вертикали соответственно.
- shear_range = 0.15 - диапазон для случайного изменения угла сдвига (искривления) изображения.
- horizontal_flip = True - случайное отражение изображения по горизонтали.
- fill_mode = “nearest” - режим заполнения для новых пикселей, возникающих при применении аугментации. Новые пиксели будут заполнены ближайшими значениями существующих пикселей.

В результате получаем модель, которая остановилась на 12 эпохах. Лучшая – 7.

Информация о модели:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 148, 148, 32)      320

 batch_normalization (BatchN  (None, 148, 148, 32)     128
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 74, 74, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 72, 72, 64)        18496

 batch_normalization_1 (Batc  (None, 72, 72, 64)       256
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 36, 36, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 34, 34, 128)       73856

 batch_normalization_2 (Batc  (None, 34, 34, 128)      512
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 17, 17, 128)      0
 2D)

 flatten (Flatten)           (None, 36992)             0

 dense (Dense)               (None, 128)               4735104

 batch_normalization_3 (Batc  (None, 128)              512
 hNormalization)

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 4,829,313
Trainable params: 4,828,609
Non-trainable params: 704
_________________________________________________________________
```

Рисунок 6 - Информация о модели 2

Выводим графики:

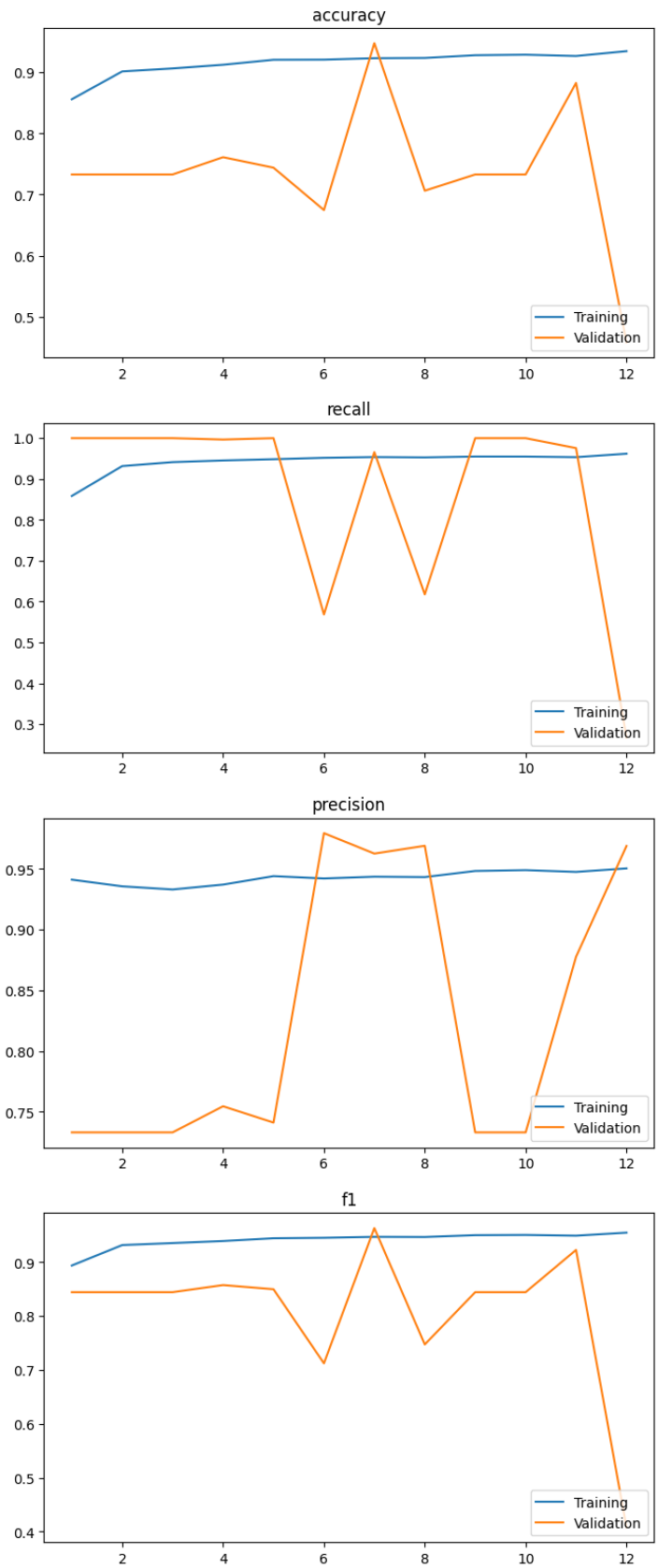

Рисунок 7 - Графики зависимости метрик от эпохи для модели 2

Оцениваем качество лучшей модели на тестовых данных:

```
Test loss: 0.1350
Test Accuracy: 0.9478
Test F1-score: 0.9632
Test recall: 0.9658
Test Precision: 0.9627
```

Добавим слой Dropout, чтобы избежать переобучения.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 148, 148, 32)      320

 batch_normalization (BatchN  (None, 148, 148, 32)     128
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 74, 74, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 72, 72, 64)        18496

 batch_normalization_1 (Batc  (None, 72, 72, 64)       256
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 36, 36, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 34, 34, 128)       73856

 batch_normalization_2 (Batc  (None, 34, 34, 128)      512
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 17, 17, 128)      0
 2D)

 flatten (Flatten)           (None, 36992)             0

 dense (Dense)               (None, 128)               4735104

 batch_normalization_3 (Batc  (None, 128)              512
 hNormalization)

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 4,829,313
Trainable params: 4,828,609
Non-trainable params: 704
_________________________________________________________________
```

Рисунок 8 - Информация о модели 3

Обучение прекратилось на 22 эпохе, минимальная функция потерь на 17 эпохе.

```
Test loss: 0.1605
Test Accuracy: 0.9418
Test F1-score: 0.9572
```

```
Test recall: 0.9705
Test Precision: 0.9459
```

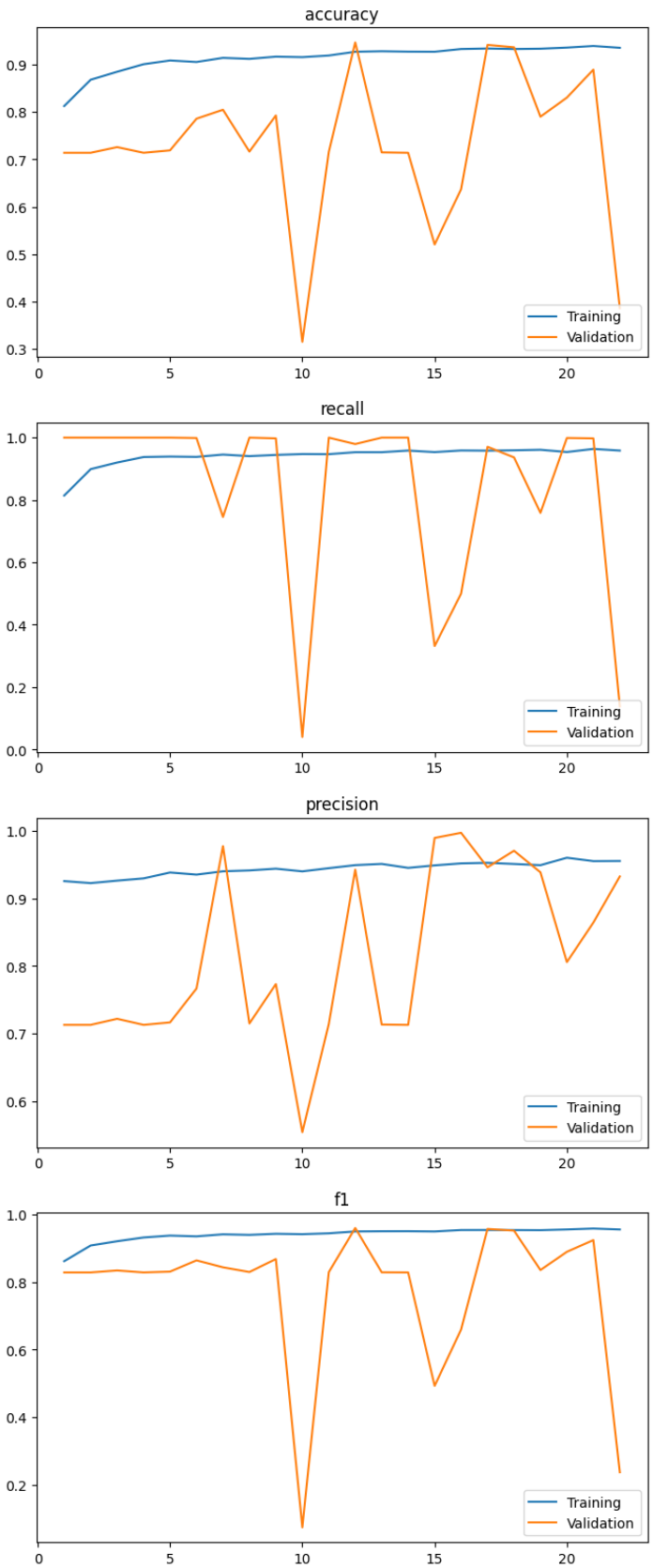

Рисунок 9 - Графики изменения метрик в зависимости от эпохи для модели 3

Построим сравнительную таблицу для 3 полученных моделей, чтобы выявить лучшую.

Таблица 1 - Сравнительная таблица трех моделей

	Базовая модель	+ BatchNormalisation + аугментация	+ Dropout
Test loss	0.1271	0.1350	0.1605
Test accuracy	0.9555	0.9478	0.9418
Test F1-score	0.9686	0.9632	0.9572
Test recall	0.9762	0.9658	0.9705
Test presicion	0.9622	0.9627	0.9459

Получаем, что лучшей по всем метрикам оказалась базовая модель.

Заключение

На основе задачи обучения модели по классификации легких на 2 класса - больные пневмонией и здоровые - были выполнены следующие задачи:

- Подготовлен датасет, состоящий из 6000 изображений различных размеров, размеченных на два класса.
- Датасет разделен на тренировочный и тестовый наборы данных с соотношением 80:20.
- Применен метод нормализации данных, использующий Rescaling, с целью масштабирования значений пикселей в диапазон от 0 до 1.
- Разработана и обучена модель нейронной сети для классификации изображений легких на 2 класса: здоровые, больные пневмонией. В качестве метрики используются recall, precision, f1-score. Обучение итоговой модели позволило достичь следующих максимальных характеристик: Test loss: 0.1271, Test Accuracy: 0.9555, Test F1-score: 0.9686, Test recall: 0.9762, Test Precision: 0.9622

Таким образом, была разработана и обучена модель нейронной сети, которая способна диагностировать заболевание пневмонией на основе рентгеновских снимков с метриками более 0,9. Это может помочь врачам в диагностике заболеваний легких и улучшить качество медицинской помощи.

Список использованных источников

[1] A CNN to Classify Pneumonia, Step by Step Using PyTorch // medium URL: https://medium.com/analytics-vidhya/a-cnn-to-classify-pneumonia-step-by-step-using-pytorch-13a90905abd7 (дата обращения: 22.04.2023).

[2] Detecting Pneumonia in X-Ray Images Using Convolutional Neural Networks // c-woo.github.io URL: https://c-woo.github.io/detecting_pneumonia_in_x-ray_images_using_convolutional_neural_networks (дата обращения: 22.04.2023).

[3] Pneumonia Diagnosis using Deep Learning: Classifying Chest X-rays with Convolutional Neural Networks // Towards Data Science URL: https://towardsdatascience.com/pneumonia-diagnosis-using-cnns-bfd71e3c05 (дата обращения: 22.02.2023).