

Беззубиков А.С.

магистрант

Астраханский государственный университет

Научный руководитель: Кошкаров А.В., к.т.н

Астраханский государственный университет

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ ДИАГНОСТИКЕ COVID-19

Аннотация

В статье произведен анализ существующих методов применения сверточных нейронных сетей для анализа рентгеновских изображений легких и анализ продуктовых решений по диагностике COVID-19 на основе снимков легких. В результате работы создана программа на языке Python, принимающая на вход рентгеновский снимок легких, обрабатывающая его подключенной функцией `check_covid` и возвращающая запрошившему json ответ результата анализа нейросети.

Ключевые слова:

COVID-19, нейронные сети, рентгеновские снимки, Keras, TensorFlow

Bezzubikov A. S.

Master's student

Astrakhan State University

Scientific supervisor: Koshkarov A.V.,

Candidate of Technical Sciences, Astrakhan State University

DEVELOPMENT OF A SOFTWARE PACKAGE TO SUPPORT DECISION-MAKING IN THE DIAGNOSIS OF COVID-19

Annotation

The article analyzes the existing methods of using convolutional neural networks for the analysis of X-ray images of the lungs and analyzes product solutions for

the diagnosis of COVID-19 based on lung images. As a result, a Python program was created that accepts an X-ray image of the lungs as input, processes it with the connected check_covid function, and returns the response of the neural network analysis result to the json requester.

Keywords:

COVID-19, neural networks, X-ray images, Keras, TensorFlow

Введение

В конце 2019 года человечество столкнулось с эпидемией—тяжелым острым респираторным синдромом коронавируса 2 (SARS CoV-2), связанной с пневмонией, называемой коронавирусной болезнью 2019 года (COVID–19),-с которой люди не ожидали столкнуться в нынешнюю эру технологий. Учитывая время, необходимое для диагностики, и финансовые затраты на лабораторные наборы, используемые для диагностики, были начаты исследования и приложения искусственного интеллекта (ИИ) и глубокого обучения для поддержки врачей, которые стремятся лечить пациентов и бороться с болезнью.

Анализ существующих разработок

Использование рентгеновских снимков для автоматического обнаружения COVID-19 может быть полезно, в частности, для стран и больниц, которые не могут приобрести лабораторный набор для проведения анализов или не имеют компьютерного томографа. Это важно, потому что в настоящее время не найдено эффективного варианта лечения, и поэтому эффективная диагностика имеет решающее значение. Использование сверточных нейронных сетей для анализа рентгеновских снимков в сочетании с возможностью быстрой загрузки данных позволит раньше диагностировать возможные проявления COVID-19 и помогать врачам в принятии решений. Непосредственно связанные с обнаружением COVID-19, три архитектуры CNN (ResNet50, InceptionV3 и InceptionResnetV2) были рассмотрены в работе [2], используя базу данных всего из 50 контрольных групп и 50 пациентов с COVID-19. Наилучшая

точность (98%) была получена с помощью ResNet50. В работе [3] семь различных глубоких моделей CNN были протестированы с использованием корпуса из 50 контрольных групп и 25 пациентов с COVID-19. Наилучшие результаты были достигнуты с моделями VGG19 и DenseNet, получив точность 0,89 и 0,91 для контроля и пациентов. Архитектура COVID-Net была предложена в работе [4].

Разработка программы

В качестве исходных изображений предлагается использовать прямую (заднюю и переднюю) проекцию рентгеновских снимков легких. Для этого использовался набор данных Kaggle по рентгеновским снимкам грудной клетки (пневмония) и отбирались 25 рентгеновских изображений здоровых пациентов. После сбора набора данных осталось 50 полных изображений, поровну разделенных на 25 изображений с подтвержденным COVID-19 и 25 изображений здоровых пациентов.

Данные рентгенографии грудной клетки коронавируса (COVID-19) находятся в каталоге dataset/, где два класса данных разделены на covid/ и normal/. Эти изображения используются для обучения модели глубокого обучения с помощью TensorFlow и Keras и автоматического предсказания, есть ли у пациента COVID-19 (то есть коронавирус). Сверточная нейронная сеть для автоматической диагностики COVID-19 была построена с использованием Keras и TensorFlow. Для проверки приходящих от клиента фотографий используется функция check_covid, находящаяся в отдельной подпрограмме.

Чтобы загрузить данные, мы захватываем все пути к изображениям в каталоге dataset. Затем для каждого пути изображения следует:

- извлечь метку класса (covid или нормальную) из пути,
- загрузить изображение и предварительно обработать его, преобразовав в порядок каналы RGB и изменить размер до 224×224 пикселей, чтобы оно было готово для нашей сверточной нейронной сети,
- обновить списки данных и меток соответственно,

- затем мы масштабируем интенсивность пикселей до диапазона [0, 1] и преобразуем данные и метки в формат массива NumPy,

Далее мы будем кодировать метки и создавать учебные/тестовые СПЛИТЫ:

```
71 lb = LabelBinarizer()
72 labels = lb.fit_transform(labels)
73 labels = to_categorical(labels) # присвоение индексов категориям
74
75 |
76 (trainX, testX, trainY, testY) = train_test_split(data, labels,
77     test_size=0.20, stratify=labels, random_state=42)
78
79
80 trainAug = ImageDataGenerator(
81     rotation_range=15,
82     fill_mode="nearest")
```

Рисунок 1 - Кодировка и создание учебных/тестовых сплитов

Каждая закодированная метка состоит из двухэлементного массива, один из элементов которого является “горячим” (т. е. 1) против “не” (т. е. 0). Затем строится разделение данных, оставляющее 80% данных для обучения и 20% для тестирования. Чтобы убедиться, что модель обучается, мы выполняем увеличение данных, устанавливая параметр случайного поворота изображения на 15 градусов по часовой стрелке или против часовой стрелки. Отсюда мы инициализируем модель VGGNet и настроим ее для тонкой настройки:

```
89 baseModel = VGG16(weights="imagenet", include_top=False,
90     input_tensor=Input(shape=(128, 128, 3)))
91
92 headModel = baseModel.output # инициализация модели
93
94 headModel = AveragePooling2D(pool_size=(4, 4))(headModel)
95
96 headModel = Flatten(name="flatten")(headModel) # входной слой нейронки
97 headModel = Dense(64, activation="relu")(headModel)
98 headModel = Dropout(0.5)(headModel) # отбрасывает ненужные нейроны
99 headModel = Dense(2, activation="softmax")(headModel)
100
101 model = Model(inputs=baseModel.input, outputs=headModel)
102 |
103 for layer in baseModel.layers:
104     layer.trainable = False
```

Рисунок 2 - Модель VGGNet

Строки 89 и 90 создают экземпляр сети VGG 16 с весами, предварительно обученными на ImageNet.

Далее используется фреймворк для веб-приложений на языке Python – Django. Архитектура Django похожа на «Модель-Представление-Контроллер» (MVC). Контроллер классической модели MVC примерно соответствует уровню, который в Django называется Представление (англ. View), а презентационная логика Представления реализуется в Django уровнем Шаблонов (англ. Template). Из-за этого уровневую архитектуру Django часто называют «Модель-Шаблон-Представление» (MTV).

Использованные источники:

1. D. Apostolopoulos and T. A. Mpesiana, “COVID-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks,” *Phys. Eng. Sci. Med.*, vol. 43, no. 2, pp. 635–640, Jun. 2020.

2. Ozsahin, I., Sekeroglu, B., Mok, G. S. P. The Use of Back Propagation Neural Networks and 18F-Florbetapir PET for Early Detection of Alzheimer’s Disease Using Alzheimer’s Disease Neuroimaging Initiative Database. *PLoS One*. 2019, 14, e0226577.

3. Narin, C. Kaya, and Z. Pamuk, “Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks,” 2020, arXiv:2003.10849. [Online]. Available: <http://arxiv.org/abs/2003.10849>

4. E. E.-D. Hemdan, M. A. Shouman, and M. E. Karar, “COVIDXNet: A framework of deep learning classifiers to diagnose COVID19 in X-ray images,” 2020, arXiv:2003.11055. [Online]. Available: <http://arxiv.org/abs/2003.11055>