«Определение вероятности заболевания COVID-19 по снимкам грудной клетки»

по дисциплине «Глубокое обучение»

Выполнили

студент гр.3530904/80105 Малинин И.И Пинаев Н.Д.

Поддубный А.А. Распереза А.Д.

Преподаватель Малеев О.Г.

Оглавление

[Постановка задачи 3](#_Toc101490052)

[Архитектура модели 3](#_Toc101490053)

[Датасет 4](#_Toc101490054)

[Реализация 4](#_Toc101490055)

[Результаты 5](#_Toc101490056)

[Веб-приложение 8](#_Toc101490057)

[Вывод 10](#_Toc101490058)

# Постановка задачи

Необходимо реализовать нейронную сеть на базе фреймворка Tensorflow для распознавания снимков рентгена грудной клетки с целью выявления заражения коронавирусной инфекцией COVID-19

# Архитектура модели

В качестве основы модели нейронной сети была взята архитектура Xception.

Модель Xception (Extreme Inception) — это современная архитектура CNN для классификации изображений, разработанная компанией Google. Она достигла выдающихся результатов в классификации изображений и превзошла Inception-V3 на наборах данных ImageNet ILSVRC и JFT. Модель Xception заняла 1-е место в конкурсе ILSVRC 2015.

Архитектура Xception CNN состоит из линейного стека слоя глубинной разделительной свертки (глубинная свертка с последующей точечной сверткой) с остаточными связями.

Обычный свёрточный слой одновременно обрабатывает как пространственную информацию (корреляцию соседних точек внутри одного канала), так и межканальную информацию, так как свёртка применяется ко всем каналам сразу. Архитектура Xception базируется на предположении о том, что эти два вида информации можно обрабатывать последовательно без потери качества работы сети, и раскладывает обычную свёртку на pointwise convolution (которая обрабатывает только межканальную корреляцию) и spatial convolution (которая обрабатывает только пространственную корреляцию в рамках отдельного канала).

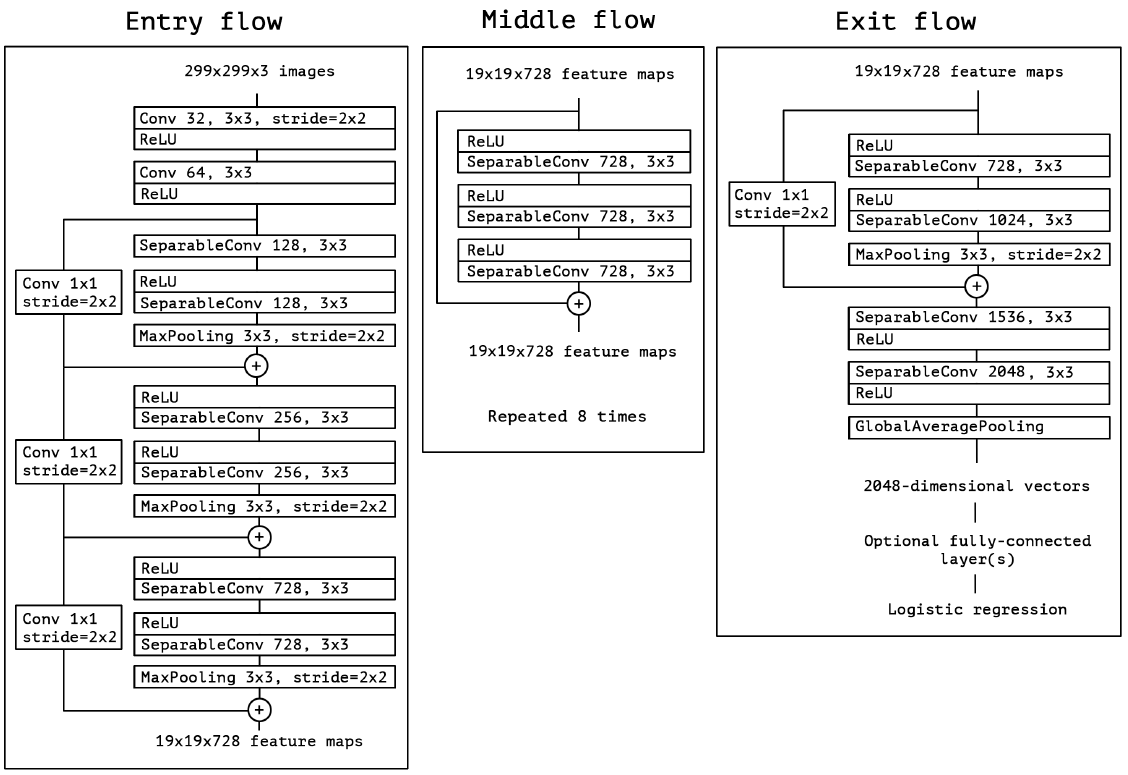


Рисунок . Общая архитектура Xception

# Датасет

В качестве датасета был взят COVID-19 Radiography Database с сайта Kaggle. Датасет состоит из 3616 снимков рентгена грудной клетки людей, больных COVID-19, 10192 снимков здоровых людей. Также датасет включает в себя снимки людей с вирусной пневмонией и с помутнениями в легких. В нашей работе используются только первые две категории снимков.

# Реализация

Снимки из датасета были перемешаны и разбиты на тренирующую, тестирующую и валидирующую выборки:

for i in range(3616):  
 if i < 2893:  
 copy\_file(listCovidPaths[i], pathCovidTrain)  
 elif i>=2893 and i<=3255:  
 copy\_file(listCovidPaths[i], pathCovidValid)  
 else :   
 copy\_file(listCovidPaths[i], pathCovidTest)  
   
for i in range(10192):  
 if i < 8154:  
 copy\_file(listNormalPaths[i], pathNormalTrain)  
 elif i>=8154 and i<=9174:  
 copy\_file(listNormalPaths[i], pathNormalValid)   
 else :   
 copy\_file(listNormalPaths[i], pathNormalTest)

Далее были созданы генераторы изображений, которые будут передавать снимки в модель с аугментацией данных:

train\_data\_gen = ImageDataGenerator(rescale=1./255 ,  
 zoom\_range= 0.3,   
 horizontal\_flip= True,   
 shear\_range= 0.2,  
 rotation\_range = 30  
 )  
  
train = train\_data\_gen.flow\_from\_directory(directory= './Data/Train',  
 class\_mode = 'binary',  
 batch\_size=64,  
 target\_size=(224,224))

Была вызвана реализация Xception из фреймворка Tensorflow

inc = tf.keras.applications.xception.Xception(  
 include\_top = False,  
 weights = 'imagenet',  
 input\_shape = (224,224,3),  
 classifier\_activation = 'softmax'  
)

Модель была дополнена слоями Flatten и Dense для последующего вывода:

x = tf.keras.layers.Flatten()(inc.output)  
prediction= tf.keras.layers.Dense(1 , activation=tf.nn.sigmoid)(x)  
  
model = tf.keras.Model(inc.input,prediction)

Далее происходит обучение модели:

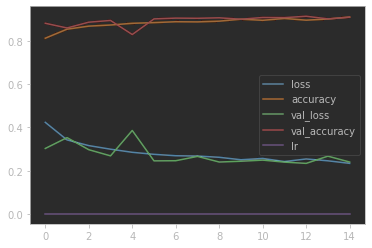
model.compile(  
 optimizer = tf.keras.optimizers.RMSprop(learning\_rate=0.0001),  
 loss = 'binary\_crossentropy',  
 metrics = ['accuracy']  
)

hist = model.fit(train, steps\_per\_epoch= len(train), epochs= 15, validation\_data= valid,class\_weight=cw,validation\_steps= len(test), callbacks=callbacks\_list)

Модель обучена за 15 эпох.

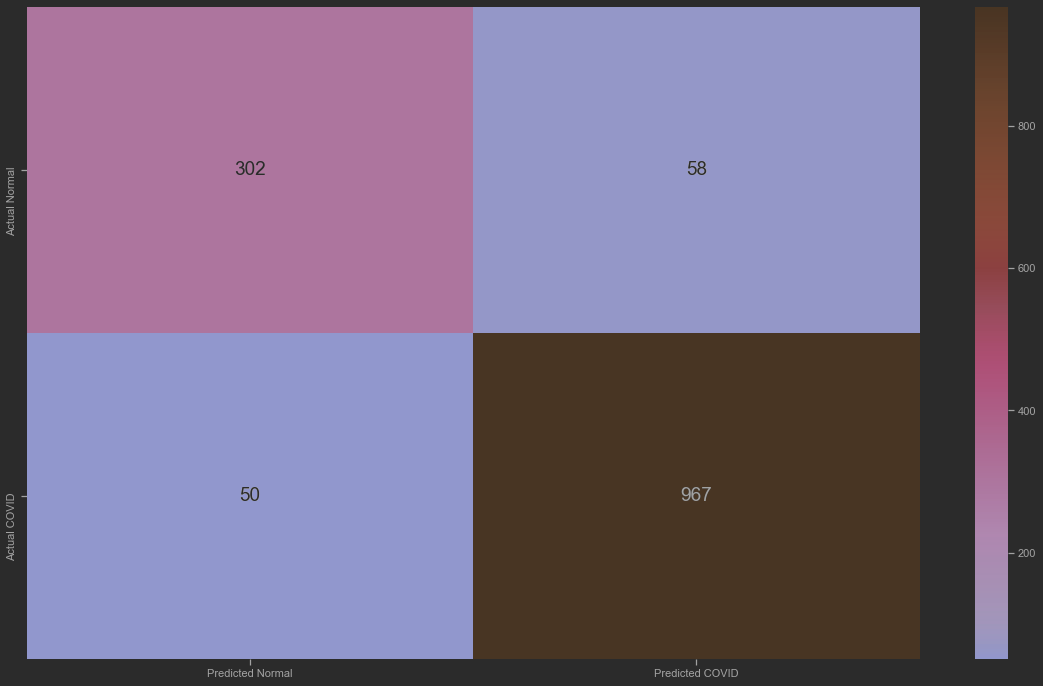
# Результаты

Графики изменения метрик (loss – функция потерь, accuracy – точность, val\_loss - функция потерь на валидации, val\_accuracy – точность на валидации, lr – learning rate) в процессе обучения:

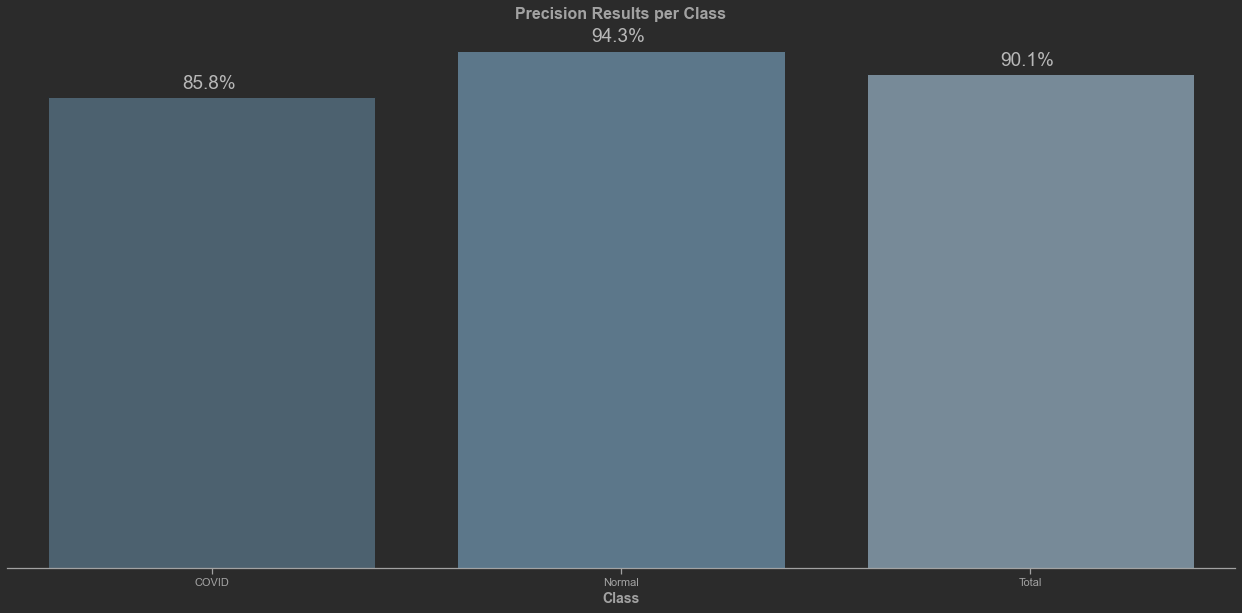
  
Изображение выглядит как текст, монитор, снимок экрана

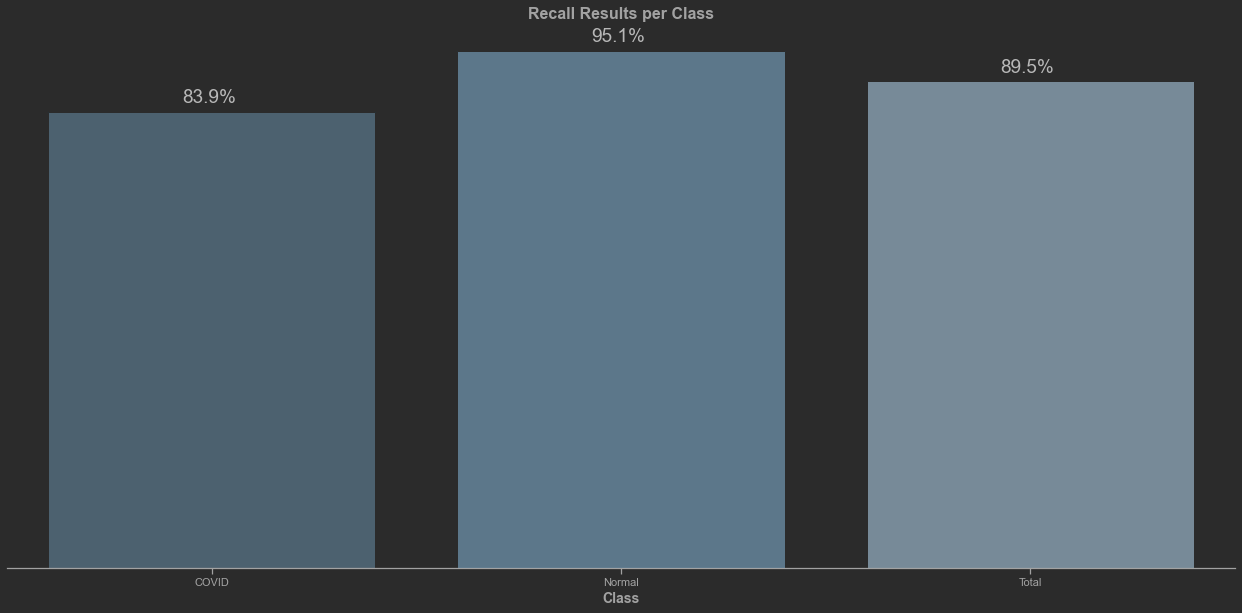
Автоматически созданное описание

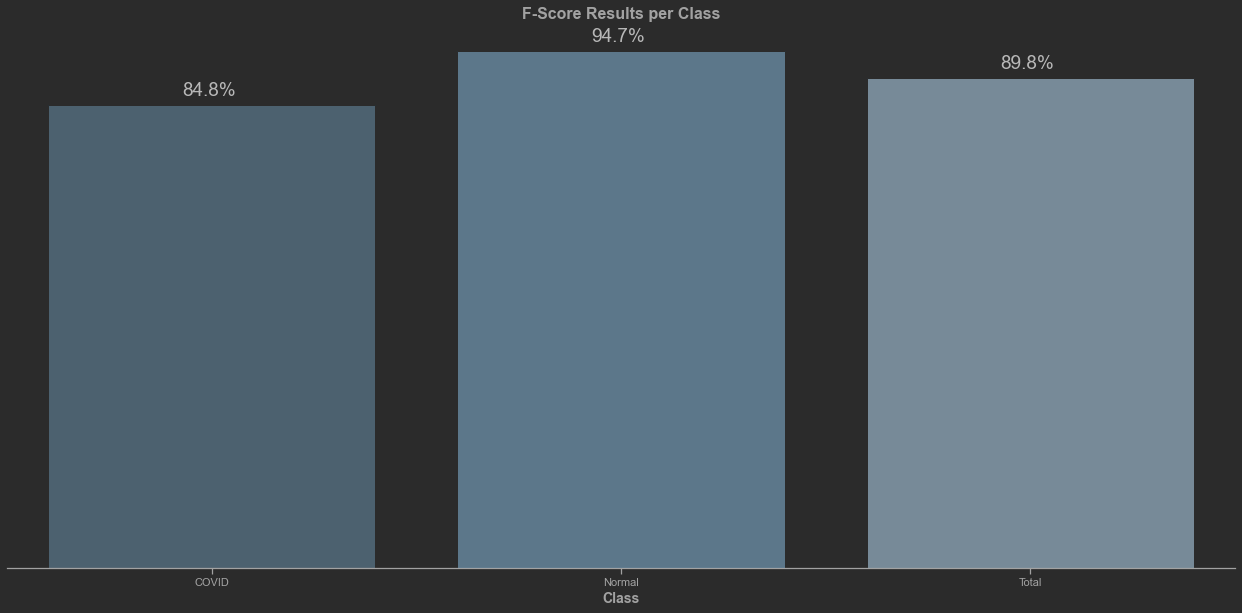
Матрица ошибок:



Гистограмма оценок классификатора (precision – точность, recall – полнота, F-score - F-мера) для модели:

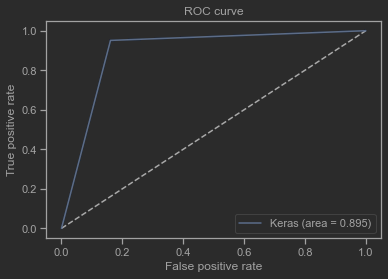






Точность модели на тестирующей выборке составила 92.15686321258545%

ROC-кривая:

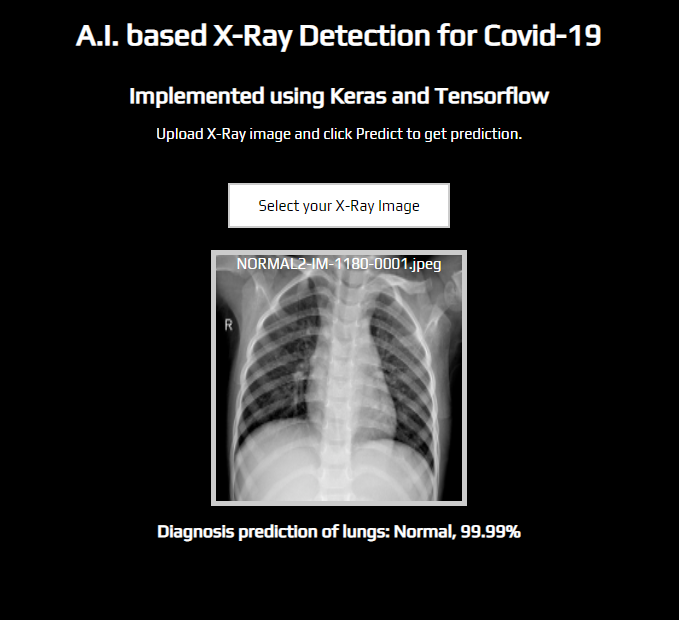


# Веб-приложение

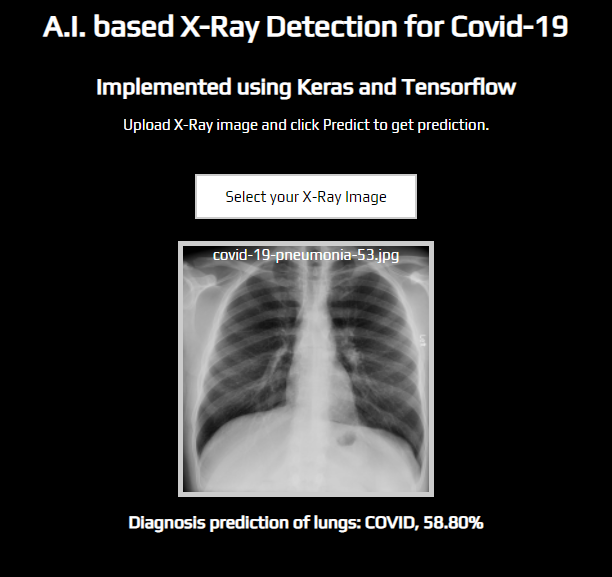
В дополнение к модели было разработано развертываемое веб-приложение чтобы продемонстрировать потенциальное применение - помощь в медицинской визуализации. Используя это приложение, пользователь может загрузить рентгеновский снимок и получить прогноз.

Веб-приложение написано с использованием фреймворка Flask.

Пример работы со случайным снимком здоровых легких, взятым из другого датасета



Пример работы со случайным снимком легких больного COVID-19, взятым из другого датасета



# Вывод

В процессе выполнения проекта была построена модель нейронной сети на основе архитектуры Xception для распознавания снимков рентгена грудной клетки с целью выявления заражения коронавирусной инфекцией COVID-19. Модель была обучена на датасете, получены метрики обучения, достигнут достаточно высокий процент точности при относительно небольшой выборке. Реализовано развертываемое веб-приложение для использования обученной модели извне.