cnn_xray_chest

July 10, 2022

0.1 Libraries

[2]:
```
import tensorflow as tf
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns
import keras
from tensorflow.keras import regularizers
from keras.models import Sequential
from keras.layers import Dense, Conv2D , MaxPool2D , Flatten , Dropout ,␣
 ↪BatchNormalization
from keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,confusion_matrix
from keras.callbacks import ReduceLROnPlateau
import cv2
```

[3]:
```
try:
    tpu = tf.distribute.cluster_resolver.TPUClusterResolver()
    print('Device:', tpu.master())
    tf.config.experimental_connect_to_cluster(tpu)
    tf.tpu.experimental.initialize_tpu_system(tpu)
    strategy = tf.distribute.experimental.TPUStrategy(tpu)
except:
    strategy = tf.distribute.get_strategy()
print('Number of replicas:', strategy.num_replicas_in_sync)

print(tf.__version__)
```

```
Number of replicas: 1
2.9.1
```

[4]:
```
import pathlib
data_dir = pathlib.Path("/Users/edoardomonnetti/Desktop/Magistrale/Reti neurali/
 ↪Project/chest_xray")
image_count = len(list(data_dir.glob('*/*/*.jpeg')))
```

```
print(image_count)
```

```
5856
```

0.2 Load data

[5]:
```
# Create the function to get the images with their label

labels = ['PNEUMONIA', 'NORMAL']
img_size = 150
def get_training_data(data_dir):
    data = []
    for label in labels:
        path = os.path.join(data_dir, label)
        class_num = labels.index(label)
        for img in os.listdir(path):
            try:
                img_arr = cv2.imread(os.path.join(path, img), cv2.
 ↪IMREAD_GRAYSCALE)
                resized_arr = cv2.resize(img_arr, (img_size, img_size)) #␣
 ↪Reshaping images to preferred size
                data.append([resized_arr, class_num])
            except Exception as e:
                print(e)
    return np.array(data, dtype=object)
```

[5]:
```
# Get the data

train_ds = get_training_data('/Users/edoardomonnetti/Desktop/Magistrale/Reti␣
 ↪neurali/Project/chest_xray/train')
test_ds = get_training_data('/Users/edoardomonnetti/Desktop/Magistrale/Reti␣
 ↪neurali/Project/chest_xray/test')
val_ds = get_training_data('/Users/edoardomonnetti/Desktop/Magistrale/Reti␣
 ↪neurali/Project/chest_xray/val')
```

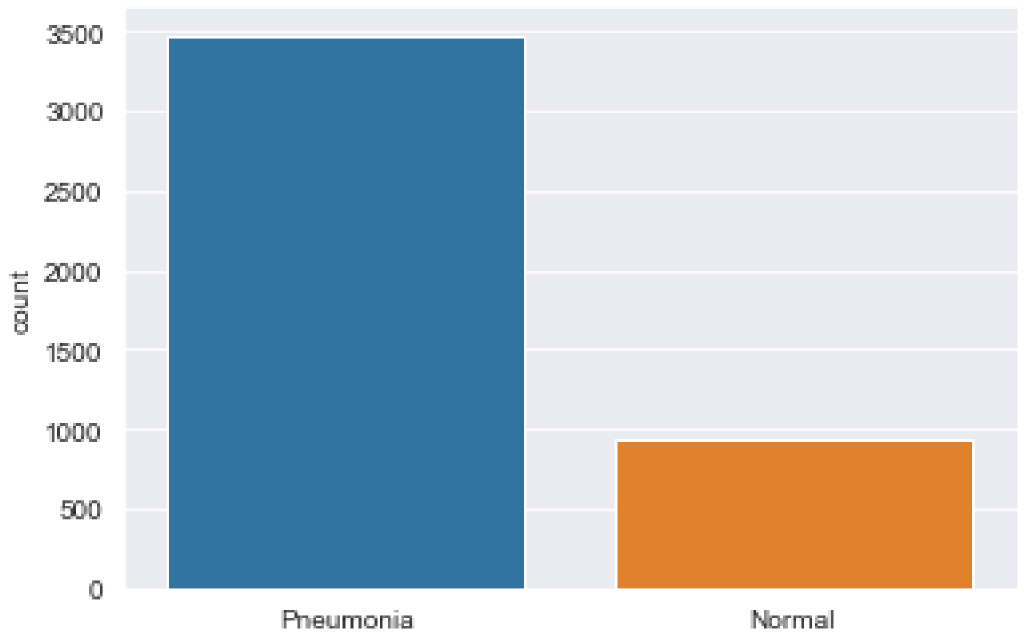
0.3 Data visualization & preprocessing

[6]:
```
train_ds.shape
```

[6]:
```
(4416, 2)
```

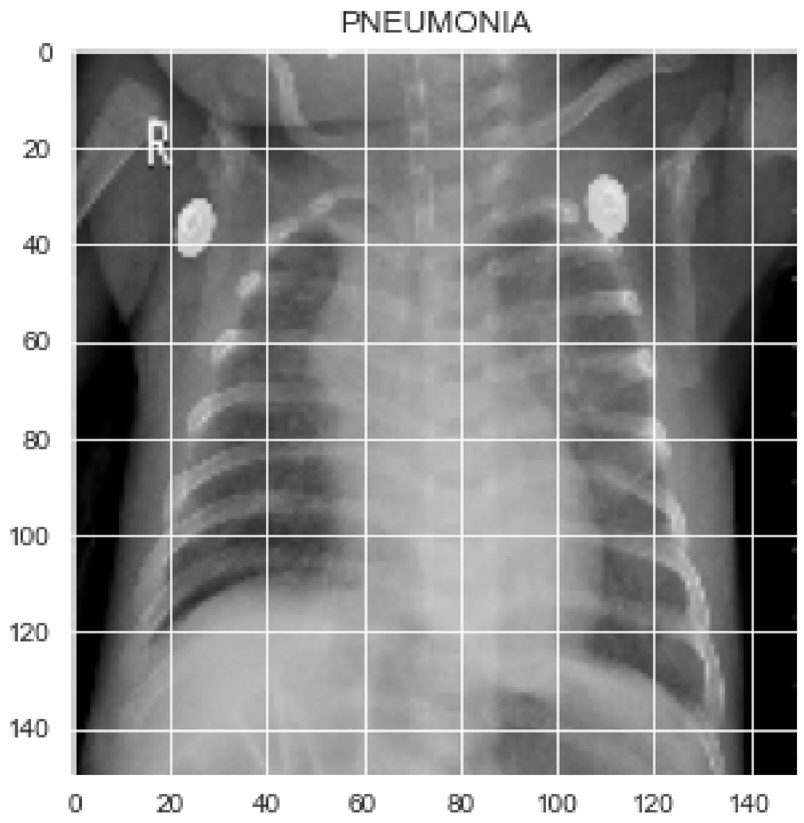
[7]:
```
# Count images for each case

l = []
for i in train_ds:
    if(i[1] == 0):
        l.append("Pneumonia")
    else:
        l.append("Normal")
sns.set_style('darkgrid')
sns.countplot(x=l)
```

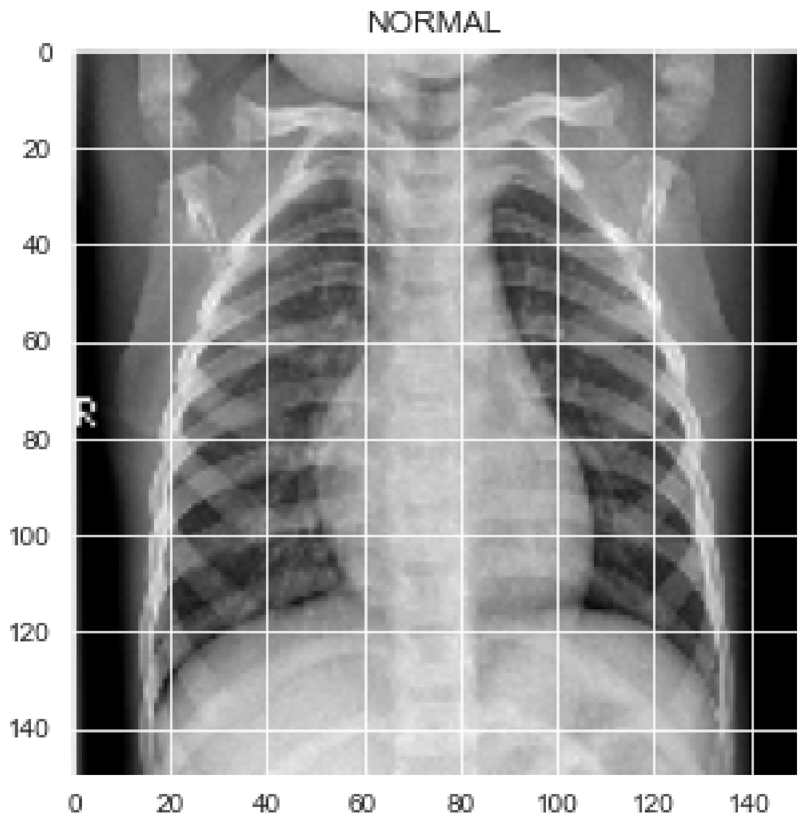
```
[7]: <AxesSubplot:ylabel='count'>
```



```
[8]: # Preview of two images

plt.figure(figsize = (5,5))
plt.imshow(train_ds[0][0], cmap='gray')
plt.title(labels[train_ds[0][1]])

plt.figure(figsize = (5,5))
plt.imshow(train_ds[-1][0], cmap='gray')
plt.title(labels[train_ds[-1][1]])
```

```
[8]: Text(0.5, 1.0, 'NORMAL')
```


PNEUMONIA
R
0
20
40
60
80
100
120
140
0
20
40
60
80
100
120
140



[9]:
```
# Create a list of the (value, label) in form of arrays

x_train = []
y_train = []

x_val = []
y_val = []

x_test = []
y_test = []

for feature, label in train_ds:
    x_train.append(feature)
    y_train.append(label)

for feature, label in test_ds:
    x_test.append(feature)
    y_test.append(label)

for feature, label in val_ds:
    x_val.append(feature)
    y_val.append(label)
```

[10]:
```
# Normalize the data

x_train = np.array(x_train) / 255.0
x_val = np.array(x_val) / 255.0
x_test = np.array(x_test) / 255.0
```

[11]:
```
x_train.shape
```

[11]:
```
(4416, 150, 150)
```

[12]:
```
# Resize data for deep learning

x_train = x_train.reshape(-1, img_size, img_size, 1)
y_train = np.array(y_train)

x_val = x_val.reshape(-1, img_size, img_size, 1)
y_val = np.array(y_val)

x_test = x_test.reshape(-1, img_size, img_size, 1)
y_test = np.array(y_test)
```

[13]:
```
x_train.shape
```

[13]:
```
(4416, 150, 150, 1)
```

0.4 Data augmentation

In order to avoid overfitting problem, we need to expand artificially our dataset. The idea is to alter the training data with small transformations to reproduce the variations.

[14]:
```
datagen = ImageDataGenerator(

        featurewise_center = False,  # set input mean to 0 over the dataset
        samplewise_center = False,  # set each sample mean to 0
        featurewise_std_normalization = False,  # divide inputs by std of the␣
 ↪dataset
        samplewise_std_normalization = False,  # divide each input by its std
        zca_whitening = False,  # apply ZCA whitening
        rotation_range = 30,  # randomly rotate images in the range (degrees, 0␣
 ↪to 180)
        zoom_range = 0.2, # Randomly zoom image
        width_shift_range = 0.1,  # randomly shift images horizontally (fraction␣
 ↪of total width)
        height_shift_range = 0.1,  # randomly shift images vertically (fraction␣
 ↪of total height)
```

```
        horizontal_flip = True,  # randomly flip images
        vertical_flip = False)  # randomly flip images


datagen.fit(x_train)
```

0.5 Model

0.5.1 CNN

[6]:
```
model = Sequential()

model.add(Conv2D(16 , (3,3) , strides = 1, activation = 'relu',
                 input_shape = (150,150,1)))
model.add(MaxPool2D())

model.add(Conv2D(32 , (3,3) , strides = 1, activation = 'relu'))
model.add(BatchNormalization())
model.add(MaxPool2D())

model.add(Conv2D(16 , (3,3) , strides = 1, activation = 'relu'))
model.add(BatchNormalization())
model.add(MaxPool2D())

#model.add(Conv2D(32 , (3,3) , strides = 1, activation = 'relu'))
#model.add(BatchNormalization())
#model.add(MaxPool2D())
model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(units = 64 , activation = 'relu',
                activity_regularizer=regularizers.L2(1e-4)
               ))
model.add(Dropout(0.3))

model.add(Dense(units = 1 , activation = 'sigmoid'))
```

[7]:
```
epochs = 100
batch_size = 32

opt = keras.optimizers.Adam()

model.compile(optimizer = opt,
              loss = tf.losses.BinaryCrossentropy(),
              metrics = ['accuracy',
                        tf.keras.metrics.Precision(name='precision'),
                        tf.keras.metrics.Recall(name='recall')
                        ])

learning_rate_reduction = ReduceLROnPlateau(monitor='val_loss', #learning rate␣
 ↪reduction
                                            patience = 2,
                                            verbose=1,
                                            factor=0.32,
                                            min_lr=0.0000001)



logdir='logs'

tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir = logdir) ␣
 ↪#savings logs

early_stopping_cb = tf.keras.callbacks.EarlyStopping(patience=10,
                                                     restore_best_weights=True)␣
 ↪#early stopping
```

[8]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 148, 148, 16)      160

 max_pooling2d (MaxPooling2D  (None, 74, 74, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 72, 72, 32)        4640

 batch_normalization (BatchN  (None, 72, 72, 32)       128
 ormalization)

 max_pooling2d_1 (MaxPooling  (None, 36, 36, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 34, 34, 16)        4624

 batch_normalization_1 (Batc  (None, 34, 34, 16)       64
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 17, 17, 16)       0
 2D)

 dropout (Dropout)           (None, 17, 17, 16)        0
 flatten (Flatten)           (None, 4624)              0

 dense (Dense)               (None, 64)                296000

 dropout_1 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 305,681
Trainable params: 305,585
Non-trainable params: 96
_________________________________________________________________
```

0.5.2 Training

[18]:
```
hist = model.fit(datagen.flow(x_train,y_train, batch_size = batch_size),
                epochs = epochs,
                validation_data = datagen.flow(x_val, y_val),
                callbacks = [tensorboard_callback,learning_rate_reduction]
                )
```

```
Epoch 1/100
138/138 [==============================] - 78s 554ms/step - loss: 0.3906 -
accuracy: 0.8519 - precision: 0.6648 - recall: 0.6153 - val_loss: 2.2628 -
val_accuracy: 0.5762 - val_precision: 0.0000e+00 - val_recall: 0.0000e+00 - lr:
0.0010
Epoch 2/100
138/138 [==============================] - 74s 533ms/step - loss: 0.3031 -
accuracy: 0.8779 - precision: 0.7243 - recall: 0.6897 - val_loss: 6.6280 -
val_accuracy: 0.5762 - val_precision: 0.0000e+00 - val_recall: 0.0000e+00 - lr:
0.0010
Epoch 3/100
138/138 [==============================] - ETA: 0s - loss: 0.2740 - accuracy:
0.8965 - precision: 0.7591 - recall: 0.7535
Epoch 3: ReduceLROnPlateau reducing learning rate to 0.00032000001519918444.
138/138 [==============================] - 68s 489ms/step - loss: 0.2740 -
accuracy: 0.8965 - precision: 0.7591 - recall: 0.7535 - val_loss: 9.9254 -
val_accuracy: 0.5762 - val_precision: 0.0000e+00 - val_recall: 0.0000e+00 - lr:
0.0010
Epoch 4/100
138/138 [==============================] - 66s 479ms/step - loss: 0.2542 -
accuracy: 0.9006 - precision: 0.7789 - recall: 0.7450 - val_loss: 0.8991 -
val_accuracy: 0.6904 - val_precision: 0.9916 - val_recall: 0.2719 - lr:
3.2000e-04
Epoch 5/100
138/138 [==============================] - 65s 465ms/step - loss: 0.2369 -
```

```
accuracy: 0.9081 - precision: 0.7879 - recall: 0.7779 - val_loss: 0.6332 -
val_accuracy: 0.6709 - val_precision: 0.5649 - val_recall: 0.9724 - lr:
3.2000e-04
Epoch 6/100
138/138 [==============================] - 61s 439ms/step - loss: 0.2334 -
accuracy: 0.9151 - precision: 0.8017 - recall: 0.7991 - val_loss: 1.0566 -
val_accuracy: 0.7148 - val_precision: 0.9931 - val_recall: 0.3295 - lr:
3.2000e-04
Epoch 7/100
138/138 [==============================] - 69s 499ms/step - loss: 0.2307 -
accuracy: 0.9169 - precision: 0.7977 - recall: 0.8172 - val_loss: 0.3036 -
val_accuracy: 0.8828 - val_precision: 0.8443 - val_recall: 0.8871 - lr:
3.2000e-04
Epoch 8/100
138/138 [==============================] - 65s 470ms/step - loss: 0.2237 -
accuracy: 0.9198 - precision: 0.8112 - recall: 0.8130 - val_loss: 0.5124 -
val_accuracy: 0.7578 - val_precision: 0.6426 - val_recall: 0.9654 - lr:
3.2000e-04
Epoch 9/100
138/138 [==============================] - ETA: 0s - loss: 0.2060 - accuracy:
0.9262 - precision: 0.8282 - recall: 0.8247
Epoch 9: ReduceLROnPlateau reducing learning rate to 0.00010240000672638416.
138/138 [==============================] - 77s 553ms/step - loss: 0.2060 -
accuracy: 0.9262 - precision: 0.8282 - recall: 0.8247 - val_loss: 4.0613 -
val_accuracy: 0.5840 - val_precision: 1.0000 - val_recall: 0.0184 - lr:
3.2000e-04
Epoch 10/100
138/138 [==============================] - 66s 477ms/step - loss: 0.2059 -
accuracy: 0.9287 - precision: 0.8417 - recall: 0.8193 - val_loss: 0.4664 -
val_accuracy: 0.8389 - val_precision: 0.9622 - val_recall: 0.6452 - lr:
1.0240e-04
Epoch 11/100
138/138 [==============================] - 61s 440ms/step - loss: 0.1991 -
accuracy: 0.9357 - precision: 0.8575 - recall: 0.8374 - val_loss: 0.2989 -
val_accuracy: 0.8906 - val_precision: 0.8889 - val_recall: 0.8479 - lr:
1.0240e-04
Epoch 12/100
138/138 [==============================] - 63s 458ms/step - loss: 0.1979 -
accuracy: 0.9334 - precision: 0.8460 - recall: 0.8406 - val_loss: 0.7978 -
val_accuracy: 0.7432 - val_precision: 0.9942 - val_recall: 0.3963 - lr:
1.0240e-04
Epoch 13/100
138/138 [==============================] - ETA: 0s - loss: 0.1873 - accuracy:
0.9337 - precision: 0.8506 - recall: 0.8353
Epoch 13: ReduceLROnPlateau reducing learning rate to 3.276800271123648e-05.
138/138 [==============================] - 75s 545ms/step - loss: 0.1873 -
accuracy: 0.9337 - precision: 0.8506 - recall: 0.8353 - val_loss: 0.3008 -
val_accuracy: 0.8857 - val_precision: 0.9182 - val_recall: 0.8018 - lr:
```

```
1.0240e-04
Epoch 14/100
138/138 [==============================] - 63s 457ms/step - loss: 0.1957 -
accuracy: 0.9350 - precision: 0.8531 - recall: 0.8395 - val_loss: 0.4200 -
val_accuracy: 0.8564 - val_precision: 0.9644 - val_recall: 0.6866 - lr:
3.2768e-05
Epoch 15/100
138/138 [==============================] - ETA: 0s - loss: 0.1953 - accuracy:
0.9389 - precision: 0.8643 - recall: 0.8459
Epoch 15: ReduceLROnPlateau reducing learning rate to 1.0485760867595673e-05.
138/138 [==============================] - 62s 448ms/step - loss: 0.1953 -
accuracy: 0.9389 - precision: 0.8643 - recall: 0.8459 - val_loss: 0.3728 -
val_accuracy: 0.8779 - val_precision: 0.9640 - val_recall: 0.7396 - lr:
3.2768e-05
Epoch 16/100
138/138 [==============================] - 61s 443ms/step - loss: 0.1912 -
accuracy: 0.9375 - precision: 0.8549 - recall: 0.8512 - val_loss: 0.3752 -
val_accuracy: 0.8691 - val_precision: 0.9545 - val_recall: 0.7258 - lr:
1.0486e-05
Epoch 17/100
138/138 [==============================] - ETA: 0s - loss: 0.1830 - accuracy:
0.9370 - precision: 0.8576 - recall: 0.8448
Epoch 17: ReduceLROnPlateau reducing learning rate to 3.3554434776306153e-06.
138/138 [==============================] - 66s 479ms/step - loss: 0.1830 -
accuracy: 0.9370 - precision: 0.8576 - recall: 0.8448 - val_loss: 0.3018 -
val_accuracy: 0.8955 - val_precision: 0.9554 - val_recall: 0.7903 - lr:
1.0486e-05
Epoch 18/100
138/138 [==============================] - 66s 473ms/step - loss: 0.1966 -
accuracy: 0.9359 - precision: 0.8515 - recall: 0.8470 - val_loss: 0.3308 -
val_accuracy: 0.8799 - val_precision: 0.9587 - val_recall: 0.7488 - lr:
3.3554e-06
Epoch 19/100
138/138 [==============================] - ETA: 0s - loss: 0.1963 - accuracy:
0.9318 - precision: 0.8397 - recall: 0.8406
Epoch 19: ReduceLROnPlateau reducing learning rate to 1.0737418779172003e-06.
138/138 [==============================] - 65s 473ms/step - loss: 0.1963 -
accuracy: 0.9318 - precision: 0.8397 - recall: 0.8406 - val_loss: 0.3728 -
val_accuracy: 0.8682 - val_precision: 0.9657 - val_recall: 0.7143 - lr:
3.3554e-06
Epoch 20/100
138/138 [==============================] - 68s 493ms/step - loss: 0.1862 -
accuracy: 0.9350 - precision: 0.8554 - recall: 0.8363 - val_loss: 0.3476 -
val_accuracy: 0.8789 - val_precision: 0.9559 - val_recall: 0.7488 - lr:
1.0737e-06
Epoch 21/100
138/138 [==============================] - ETA: 0s - loss: 0.1965 - accuracy:
0.9300 - precision: 0.8398 - recall: 0.8300
```

```
Epoch 21: ReduceLROnPlateau reducing learning rate to 3.4359738492639736e-07.
138/138 [==============================] - 67s 488ms/step - loss: 0.1965 -
accuracy: 0.9300 - precision: 0.8398 - recall: 0.8300 - val_loss: 0.3509 -
val_accuracy: 0.8750 - val_precision: 0.9636 - val_recall: 0.7327 - lr:
1.0737e-06
Epoch 22/100
138/138 [==============================] - 65s 472ms/step - loss: 0.1799 -
accuracy: 0.9389 - precision: 0.8643 - recall: 0.8459 - val_loss: 0.3566 -
val_accuracy: 0.8730 - val_precision: 0.9720 - val_recall: 0.7212 - lr:
3.4360e-07
Epoch 23/100
138/138 [==============================] - ETA: 0s - loss: 0.1867 - accuracy:
0.9395 - precision: 0.8703 - recall: 0.8417
Epoch 23: ReduceLROnPlateau reducing learning rate to 1.0995116099365987e-07.
138/138 [==============================] - 63s 459ms/step - loss: 0.1867 -
accuracy: 0.9395 - precision: 0.8703 - recall: 0.8417 - val_loss: 0.3638 -
val_accuracy: 0.8711 - val_precision: 0.9660 - val_recall: 0.7212 - lr:
3.4360e-07
Epoch 24/100
138/138 [==============================] - 67s 484ms/step - loss: 0.1907 -
accuracy: 0.9386 - precision: 0.8602 - recall: 0.8502 - val_loss: 0.3715 -
val_accuracy: 0.8672 - val_precision: 0.9488 - val_recall: 0.7258 - lr:
1.0995e-07
Epoch 25/100
138/138 [==============================] - ETA: 0s - loss: 0.1867 - accuracy:
0.9350 - precision: 0.8501 - recall: 0.8438
Epoch 25: ReduceLROnPlateau reducing learning rate to 1e-07.
138/138 [==============================] - 63s 455ms/step - loss: 0.1867 -
accuracy: 0.9350 - precision: 0.8501 - recall: 0.8438 - val_loss: 0.3631 -
val_accuracy: 0.8838 - val_precision: 0.9730 - val_recall: 0.7465 - lr:
1.0995e-07
Epoch 26/100
138/138 [==============================] - 65s 471ms/step - loss: 0.1871 -
accuracy: 0.9359 - precision: 0.8522 - recall: 0.8459 - val_loss: 0.3528 -
val_accuracy: 0.8750 - val_precision: 0.9581 - val_recall: 0.7373 - lr:
1.0000e-07
Epoch 27/100
138/138 [==============================] - 63s 459ms/step - loss: 0.1882 -
accuracy: 0.9395 - precision: 0.8624 - recall: 0.8523 - val_loss: 0.3578 -
val_accuracy: 0.8662 - val_precision: 0.9569 - val_recall: 0.7166 - lr:
1.0000e-07
Epoch 28/100
138/138 [==============================] - 60s 436ms/step - loss: 0.1862 -
accuracy: 0.9359 - precision: 0.8507 - recall: 0.8480 - val_loss: 0.3515 -
val_accuracy: 0.8828 - val_precision: 0.9816 - val_recall: 0.7373 - lr:
1.0000e-07
Epoch 29/100
138/138 [==============================] - 62s 450ms/step - loss: 0.1933 -
```

```
accuracy: 0.9337 - precision: 0.8506 - recall: 0.8353 - val_loss: 0.3466 -
val_accuracy: 0.8740 - val_precision: 0.9635 - val_recall: 0.7304 - lr:
1.0000e-07
Epoch 30/100
138/138 [==============================] - 66s 480ms/step - loss: 0.1925 -
accuracy: 0.9382 - precision: 0.8623 - recall: 0.8448 - val_loss: 0.3817 -
val_accuracy: 0.8711 - val_precision: 0.9604 - val_recall: 0.7258 - lr:
1.0000e-07
Epoch 31/100
138/138 [==============================] - 67s 485ms/step - loss: 0.1880 -
accuracy: 0.9395 - precision: 0.8608 - recall: 0.8544 - val_loss: 0.3503 -
val_accuracy: 0.8730 - val_precision: 0.9663 - val_recall: 0.7258 - lr:
1.0000e-07
Epoch 32/100
138/138 [==============================] - 66s 477ms/step - loss: 0.1933 -
accuracy: 0.9382 - precision: 0.8623 - recall: 0.8448 - val_loss: 0.3658 -
val_accuracy: 0.8750 - val_precision: 0.9722 - val_recall: 0.7258 - lr:
1.0000e-07
Epoch 33/100
138/138 [==============================] - 67s 479ms/step - loss: 0.1964 -
accuracy: 0.9339 - precision: 0.8501 - recall: 0.8374 - val_loss: 0.3579 -
val_accuracy: 0.8760 - val_precision: 0.9666 - val_recall: 0.7327 - lr:
1.0000e-07
Epoch 34/100
138/138 [==============================] - 66s 480ms/step - loss: 0.1836 -
accuracy: 0.9370 - precision: 0.8568 - recall: 0.8459 - val_loss: 0.3569 -
val_accuracy: 0.8779 - val_precision: 0.9668 - val_recall: 0.7373 - lr:
1.0000e-07
Epoch 35/100
138/138 [==============================] - 67s 487ms/step - loss: 0.1886 -
accuracy: 0.9364 - precision: 0.8541 - recall: 0.8459 - val_loss: 0.3541 -
val_accuracy: 0.8760 - val_precision: 0.9694 - val_recall: 0.7304 - lr:
1.0000e-07
Epoch 36/100
138/138 [==============================] - 65s 470ms/step - loss: 0.2011 -
accuracy: 0.9330 - precision: 0.8435 - recall: 0.8417 - val_loss: 0.3575 -
val_accuracy: 0.8789 - val_precision: 0.9726 - val_recall: 0.7350 - lr:
1.0000e-07
Epoch 37/100
138/138 [==============================] - 59s 428ms/step - loss: 0.1935 -
accuracy: 0.9334 - precision: 0.8402 - recall: 0.8491 - val_loss: 0.3621 -
val_accuracy: 0.8701 - val_precision: 0.9631 - val_recall: 0.7212 - lr:
1.0000e-07
Epoch 38/100
138/138 [==============================] - 60s 432ms/step - loss: 0.1893 -
accuracy: 0.9377 - precision: 0.8733 - recall: 0.8278 - val_loss: 0.3712 -
val_accuracy: 0.8643 - val_precision: 0.9624 - val_recall: 0.7074 - lr:
1.0000e-07
```

```
Epoch 39/100
138/138 [==============================] - 59s 430ms/step - loss: 0.1856 -
accuracy: 0.9386 - precision: 0.8633 - recall: 0.8459 - val_loss: 0.3535 -
val_accuracy: 0.8730 - val_precision: 0.9606 - val_recall: 0.7304 - lr:
1.0000e-07
Epoch 40/100
138/138 [==============================] - 60s 433ms/step - loss: 0.1918 -
accuracy: 0.9350 - precision: 0.8562 - recall: 0.8353 - val_loss: 0.3647 -
val_accuracy: 0.8633 - val_precision: 0.9652 - val_recall: 0.7028 - lr:
1.0000e-07
Epoch 41/100
138/138 [==============================] - 60s 433ms/step - loss: 0.1862 -
accuracy: 0.9391 - precision: 0.8621 - recall: 0.8502 - val_loss: 0.3765 -
val_accuracy: 0.8574 - val_precision: 0.9444 - val_recall: 0.7051 - lr:
1.0000e-07
Epoch 42/100
138/138 [==============================] - 61s 440ms/step - loss: 0.1872 -
accuracy: 0.9361 - precision: 0.8547 - recall: 0.8438 - val_loss: 0.3613 -
val_accuracy: 0.8750 - val_precision: 0.9693 - val_recall: 0.7281 - lr:
1.0000e-07
Epoch 43/100
138/138 [==============================] - 70s 504ms/step - loss: 0.1851 -
accuracy: 0.9389 - precision: 0.8581 - recall: 0.8544 - val_loss: 0.3899 -
val_accuracy: 0.8584 - val_precision: 0.9474 - val_recall: 0.7051 - lr:
1.0000e-07
Epoch 44/100
138/138 [==============================] - 73s 530ms/step - loss: 0.1909 -
accuracy: 0.9364 - precision: 0.8511 - recall: 0.8502 - val_loss: 0.3635 -
val_accuracy: 0.8672 - val_precision: 0.9599 - val_recall: 0.7166 - lr:
1.0000e-07
Epoch 45/100
138/138 [==============================] - 60s 435ms/step - loss: 0.1855 -
accuracy: 0.9407 - precision: 0.8600 - recall: 0.8618 - val_loss: 0.3700 -
val_accuracy: 0.8682 - val_precision: 0.9657 - val_recall: 0.7143 - lr:
1.0000e-07
Epoch 46/100
138/138 [==============================] - 60s 434ms/step - loss: 0.1905 -
accuracy: 0.9321 - precision: 0.8363 - recall: 0.8470 - val_loss: 0.3861 -
val_accuracy: 0.8613 - val_precision: 0.9591 - val_recall: 0.7028 - lr:
1.0000e-07
Epoch 47/100
138/138 [==============================] - 60s 433ms/step - loss: 0.1885 -
accuracy: 0.9418 - precision: 0.8677 - recall: 0.8576 - val_loss: 0.3598 -
val_accuracy: 0.8750 - val_precision: 0.9722 - val_recall: 0.7258 - lr:
1.0000e-07
Epoch 48/100
138/138 [==============================] - 60s 434ms/step - loss: 0.1881 -
accuracy: 0.9321 - precision: 0.8503 - recall: 0.8268 - val_loss: 0.3578 -
```

```
val_accuracy: 0.8594 - val_precision: 0.9677 - val_recall: 0.6912 - lr:
1.0000e-07
Epoch 49/100
138/138 [==============================] - 60s 434ms/step - loss: 0.1870 -
accuracy: 0.9382 - precision: 0.8538 - recall: 0.8565 - val_loss: 0.3635 -
val_accuracy: 0.8662 - val_precision: 0.9685 - val_recall: 0.7074 - lr:
1.0000e-07
Epoch 50/100
138/138 [==============================] - 60s 433ms/step - loss: 0.1932 -
accuracy: 0.9316 - precision: 0.8353 - recall: 0.8459 - val_loss: 0.3468 -
val_accuracy: 0.8662 - val_precision: 0.9459 - val_recall: 0.7258 - lr:
1.0000e-07
Epoch 51/100
138/138 [==============================] - 60s 434ms/step - loss: 0.1853 -
accuracy: 0.9393 - precision: 0.8654 - recall: 0.8470 - val_loss: 0.3523 -
val_accuracy: 0.8711 - val_precision: 0.9689 - val_recall: 0.7189 - lr:
1.0000e-07
Epoch 52/100
138/138 [==============================] - 60s 433ms/step - loss: 0.1879 -
accuracy: 0.9382 - precision: 0.8599 - recall: 0.8480 - val_loss: 0.3518 -
val_accuracy: 0.8760 - val_precision: 0.9666 - val_recall: 0.7327 - lr:
1.0000e-07
Epoch 53/100
138/138 [==============================] - 60s 433ms/step - loss: 0.1889 -
accuracy: 0.9389 - precision: 0.8498 - recall: 0.8661 - val_loss: 0.3665 -
val_accuracy: 0.8633 - val_precision: 0.9652 - val_recall: 0.7028 - lr:
1.0000e-07
Epoch 54/100
138/138 [==============================] - 71s 515ms/step - loss: 0.1983 -
accuracy: 0.9339 - precision: 0.8516 - recall: 0.8353 - val_loss: 0.3545 -
val_accuracy: 0.8633 - val_precision: 0.9565 - val_recall: 0.7097 - lr:
1.0000e-07
Epoch 55/100
138/138 [==============================] - 65s 473ms/step - loss: 0.1850 -
accuracy: 0.9380 - precision: 0.8598 - recall: 0.8470 - val_loss: 0.3729 -
val_accuracy: 0.8604 - val_precision: 0.9533 - val_recall: 0.7051 - lr:
1.0000e-07
Epoch 56/100
138/138 [==============================] - 68s 493ms/step - loss: 0.1861 -
accuracy: 0.9398 - precision: 0.8617 - recall: 0.8544 - val_loss: 0.3647 -
val_accuracy: 0.8730 - val_precision: 0.9634 - val_recall: 0.7281 - lr:
1.0000e-07
Epoch 57/100
138/138 [==============================] - 73s 527ms/step - loss: 0.1766 -
accuracy: 0.9427 - precision: 0.8780 - recall: 0.8491 - val_loss: 0.3669 -
val_accuracy: 0.8652 - val_precision: 0.9568 - val_recall: 0.7143 - lr:
1.0000e-07
Epoch 58/100
```

```
138/138 [==============================] - 75s 542ms/step - loss: 0.1952 -
accuracy: 0.9314 - precision: 0.8394 - recall: 0.8385 - val_loss: 0.3647 -
val_accuracy: 0.8730 - val_precision: 0.9497 - val_recall: 0.7396 - lr:
1.0000e-07
Epoch 59/100
138/138 [==============================] - 67s 482ms/step - loss: 0.1813 -
accuracy: 0.9375 - precision: 0.8571 - recall: 0.8480 - val_loss: 0.3827 -
val_accuracy: 0.8818 - val_precision: 0.9728 - val_recall: 0.7419 - lr:
1.0000e-07
Epoch 60/100
138/138 [==============================] - 68s 490ms/step - loss: 0.1914 -
accuracy: 0.9364 - precision: 0.8511 - recall: 0.8502 - val_loss: 0.3711 -
val_accuracy: 0.8730 - val_precision: 0.9691 - val_recall: 0.7235 - lr:
1.0000e-07
Epoch 61/100
138/138 [==============================] - 63s 456ms/step - loss: 0.1816 -
accuracy: 0.9370 - precision: 0.8545 - recall: 0.8491 - val_loss: 0.3729 -
val_accuracy: 0.8672 - val_precision: 0.9571 - val_recall: 0.7189 - lr:
1.0000e-07
Epoch 62/100
138/138 [==============================] - 64s 463ms/step - loss: 0.1774 -
accuracy: 0.9434 - precision: 0.8719 - recall: 0.8608 - val_loss: 0.3551 -
val_accuracy: 0.8721 - val_precision: 0.9779 - val_recall: 0.7143 - lr:
1.0000e-07
Epoch 63/100
138/138 [==============================] - 75s 540ms/step - loss: 0.1959 -
accuracy: 0.9364 - precision: 0.8548 - recall: 0.8448 - val_loss: 0.3730 -
val_accuracy: 0.8672 - val_precision: 0.9686 - val_recall: 0.7097 - lr:
1.0000e-07
Epoch 64/100
138/138 [==============================] - 70s 509ms/step - loss: 0.1956 -
accuracy: 0.9346 - precision: 0.8490 - recall: 0.8427 - val_loss: 0.3579 -
val_accuracy: 0.8701 - val_precision: 0.9574 - val_recall: 0.7258 - lr:
1.0000e-07
Epoch 65/100
138/138 [==============================] - 71s 513ms/step - loss: 0.1858 -
accuracy: 0.9411 - precision: 0.8657 - recall: 0.8565 - val_loss: 0.3579 -
val_accuracy: 0.8682 - val_precision: 0.9628 - val_recall: 0.7166 - lr:
1.0000e-07
Epoch 66/100
138/138 [==============================] - 74s 531ms/step - loss: 0.1964 -
accuracy: 0.9339 - precision: 0.8478 - recall: 0.8406 - val_loss: 0.3674 -
val_accuracy: 0.8711 - val_precision: 0.9576 - val_recall: 0.7281 - lr:
1.0000e-07
Epoch 67/100
138/138 [==============================] - 74s 539ms/step - loss: 0.1884 -
accuracy: 0.9327 - precision: 0.8586 - recall: 0.8193 - val_loss: 0.3522 -
val_accuracy: 0.8701 - val_precision: 0.9688 - val_recall: 0.7166 - lr:
```

```
1.0000e-07
Epoch 68/100
138/138 [==============================] - 74s 536ms/step - loss: 0.1956 -
accuracy: 0.9352 - precision: 0.8473 - recall: 0.8491 - val_loss: 0.3853 -
val_accuracy: 0.8633 - val_precision: 0.9652 - val_recall: 0.7028 - lr:
1.0000e-07
Epoch 69/100
138/138 [==============================] - 81s 588ms/step - loss: 0.1882 -
accuracy: 0.9359 - precision: 0.8538 - recall: 0.8438 - val_loss: 0.3551 -
val_accuracy: 0.8604 - val_precision: 0.9477 - val_recall: 0.7097 - lr:
1.0000e-07
Epoch 70/100
138/138 [==============================] - 74s 535ms/step - loss: 0.1869 -
accuracy: 0.9364 - precision: 0.8564 - recall: 0.8427 - val_loss: 0.3705 -
val_accuracy: 0.8750 - val_precision: 0.9554 - val_recall: 0.7396 - lr:
1.0000e-07
Epoch 71/100
138/138 [==============================] - 61s 440ms/step - loss: 0.1886 -
accuracy: 0.9373 - precision: 0.8524 - recall: 0.8533 - val_loss: 0.3763 -
val_accuracy: 0.8594 - val_precision: 0.9531 - val_recall: 0.7028 - lr:
1.0000e-07
Epoch 72/100
138/138 [==============================] - 68s 495ms/step - loss: 0.1982 -
accuracy: 0.9359 - precision: 0.8530 - recall: 0.8448 - val_loss: 0.3508 -
val_accuracy: 0.8682 - val_precision: 0.9572 - val_recall: 0.7212 - lr:
1.0000e-07
Epoch 73/100
138/138 [==============================] - 74s 538ms/step - loss: 0.1883 -
accuracy: 0.9327 - precision: 0.8470 - recall: 0.8353 - val_loss: 0.3856 -
val_accuracy: 0.8672 - val_precision: 0.9656 - val_recall: 0.7120 - lr:
1.0000e-07
Epoch 74/100
138/138 [==============================] - 69s 496ms/step - loss: 0.1900 -
accuracy: 0.9395 - precision: 0.8631 - recall: 0.8512 - val_loss: 0.3598 -
val_accuracy: 0.8730 - val_precision: 0.9524 - val_recall: 0.7373 - lr:
1.0000e-07
Epoch 75/100
138/138 [==============================] - 73s 527ms/step - loss: 0.1865 -
accuracy: 0.9384 - precision: 0.8570 - recall: 0.8533 - val_loss: 0.3627 -
val_accuracy: 0.8760 - val_precision: 0.9752 - val_recall: 0.7258 - lr:
1.0000e-07
Epoch 76/100
138/138 [==============================] - 66s 479ms/step - loss: 0.1897 -
accuracy: 0.9436 - precision: 0.8802 - recall: 0.8512 - val_loss: 0.3676 -
val_accuracy: 0.8770 - val_precision: 0.9843 - val_recall: 0.7212 - lr:
1.0000e-07
Epoch 77/100
138/138 [==============================] - 76s 551ms/step - loss: 0.1833 -
```

```
accuracy: 0.9373 - precision: 0.8555 - recall: 0.8491 - val_loss: 0.3405 -
val_accuracy: 0.8721 - val_precision: 0.9605 - val_recall: 0.7281 - lr:
1.0000e-07
Epoch 78/100
138/138 [==============================] - 69s 497ms/step - loss: 0.1803 -
accuracy: 0.9373 - precision: 0.8547 - recall: 0.8502 - val_loss: 0.3609 -
val_accuracy: 0.8691 - val_precision: 0.9717 - val_recall: 0.7120 - lr:
1.0000e-07
Epoch 79/100
138/138 [==============================] - 67s 486ms/step - loss: 0.1873 -
accuracy: 0.9352 - precision: 0.8518 - recall: 0.8427 - val_loss: 0.3748 -
val_accuracy: 0.8662 - val_precision: 0.9714 - val_recall: 0.7051 - lr:
1.0000e-07
Epoch 80/100
138/138 [==============================] - 65s 473ms/step - loss: 0.1815 -
accuracy: 0.9352 - precision: 0.8595 - recall: 0.8321 - val_loss: 0.3706 -
val_accuracy: 0.8564 - val_precision: 0.9644 - val_recall: 0.6866 - lr:
1.0000e-07
Epoch 81/100
138/138 [==============================] - 76s 546ms/step - loss: 0.1893 -
accuracy: 0.9375 - precision: 0.8682 - recall: 0.8332 - val_loss: 0.3697 -
val_accuracy: 0.8584 - val_precision: 0.9617 - val_recall: 0.6935 - lr:
1.0000e-07
Epoch 82/100
138/138 [==============================] - 81s 589ms/step - loss: 0.1883 -
accuracy: 0.9382 - precision: 0.8576 - recall: 0.8512 - val_loss: 0.3391 -
val_accuracy: 0.8633 - val_precision: 0.9482 - val_recall: 0.7166 - lr:
1.0000e-07
Epoch 83/100
138/138 [==============================] - 68s 490ms/step - loss: 0.1786 -
accuracy: 0.9432 - precision: 0.8726 - recall: 0.8587 - val_loss: 0.3342 -
val_accuracy: 0.8691 - val_precision: 0.9688 - val_recall: 0.7143 - lr:
1.0000e-07
Epoch 84/100
138/138 [==============================] - 73s 526ms/step - loss: 0.1994 -
accuracy: 0.9348 - precision: 0.8545 - recall: 0.8363 - val_loss: 0.3710 -
val_accuracy: 0.8701 - val_precision: 0.9493 - val_recall: 0.7327 - lr:
1.0000e-07
Epoch 85/100
138/138 [==============================] - 70s 510ms/step - loss: 0.1910 -
accuracy: 0.9346 - precision: 0.8490 - recall: 0.8427 - val_loss: 0.3524 -
val_accuracy: 0.8711 - val_precision: 0.9604 - val_recall: 0.7258 - lr:
1.0000e-07
Epoch 86/100
138/138 [==============================] - 65s 473ms/step - loss: 0.1934 -
accuracy: 0.9312 - precision: 0.8443 - recall: 0.8300 - val_loss: 0.3539 -
val_accuracy: 0.8760 - val_precision: 0.9610 - val_recall: 0.7373 - lr:
1.0000e-07
```

```
Epoch 87/100
138/138 [==============================] - 73s 531ms/step - loss: 0.1845 -
accuracy: 0.9380 - precision: 0.8590 - recall: 0.8480 - val_loss: 0.3677 -
val_accuracy: 0.8770 - val_precision: 0.9611 - val_recall: 0.7396 - lr:
1.0000e-07
Epoch 88/100
138/138 [==============================] - 65s 472ms/step - loss: 0.1849 -
accuracy: 0.9346 - precision: 0.8543 - recall: 0.8353 - val_loss: 0.3781 -
val_accuracy: 0.8633 - val_precision: 0.9682 - val_recall: 0.7005 - lr:
1.0000e-07
Epoch 89/100
138/138 [==============================] - 71s 512ms/step - loss: 0.1887 -
accuracy: 0.9341 - precision: 0.8587 - recall: 0.8268 - val_loss: 0.3606 -
val_accuracy: 0.8711 - val_precision: 0.9604 - val_recall: 0.7258 - lr:
1.0000e-07
Epoch 90/100
138/138 [==============================] - 69s 498ms/step - loss: 0.1967 -
accuracy: 0.9341 - precision: 0.8525 - recall: 0.8353 - val_loss: 0.3560 -
val_accuracy: 0.8711 - val_precision: 0.9719 - val_recall: 0.7166 - lr:
1.0000e-07
Epoch 91/100
138/138 [==============================] - 67s 488ms/step - loss: 0.1814 -
accuracy: 0.9427 - precision: 0.8739 - recall: 0.8544 - val_loss: 0.3802 -
val_accuracy: 0.8555 - val_precision: 0.9643 - val_recall: 0.6843 - lr:
1.0000e-07
Epoch 92/100
138/138 [==============================] - 65s 470ms/step - loss: 0.1866 -
accuracy: 0.9364 - precision: 0.8541 - recall: 0.8459 - val_loss: 0.3689 -
val_accuracy: 0.8643 - val_precision: 0.9538 - val_recall: 0.7143 - lr:
1.0000e-07
Epoch 93/100
138/138 [==============================] - 60s 435ms/step - loss: 0.1869 -
accuracy: 0.9352 - precision: 0.8458 - recall: 0.8512 - val_loss: 0.3436 -
val_accuracy: 0.8828 - val_precision: 0.9673 - val_recall: 0.7488 - lr:
1.0000e-07
Epoch 94/100
138/138 [==============================] - 60s 434ms/step - loss: 0.1908 -
accuracy: 0.9395 - precision: 0.8671 - recall: 0.8459 - val_loss: 0.3701 -
val_accuracy: 0.8662 - val_precision: 0.9598 - val_recall: 0.7143 - lr:
1.0000e-07
Epoch 95/100
138/138 [==============================] - 60s 434ms/step - loss: 0.1846 -
accuracy: 0.9389 - precision: 0.8604 - recall: 0.8512 - val_loss: 0.3849 -
val_accuracy: 0.8623 - val_precision: 0.9592 - val_recall: 0.7051 - lr:
1.0000e-07
Epoch 96/100
138/138 [==============================] - 61s 438ms/step - loss: 0.1815 -
accuracy: 0.9377 - precision: 0.8520 - recall: 0.8565 - val_loss: 0.3602 -
```

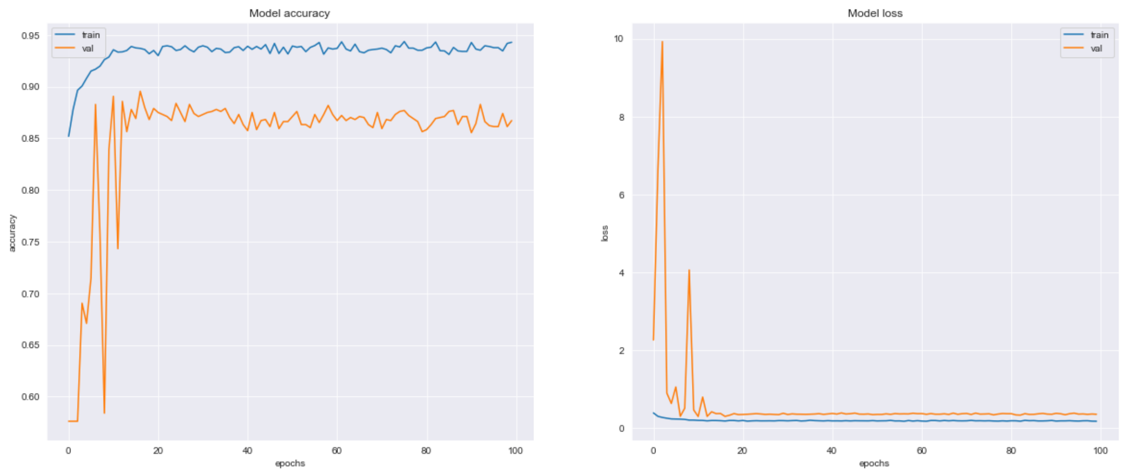
```
val_accuracy: 0.8613 - val_precision: 0.9506 - val_recall: 0.7097 - lr:
1.0000e-07
Epoch 97/100
138/138 [==============================] - 60s 434ms/step - loss: 0.1885 -
accuracy: 0.9377 - precision: 0.8535 - recall: 0.8544 - val_loss: 0.3640 -
val_accuracy: 0.8613 - val_precision: 0.9534 - val_recall: 0.7074 - lr:
1.0000e-07
Epoch 98/100
138/138 [==============================] - 60s 437ms/step - loss: 0.1903 -
accuracy: 0.9346 - precision: 0.8575 - recall: 0.8310 - val_loss: 0.3540 -
val_accuracy: 0.8740 - val_precision: 0.9751 - val_recall: 0.7212 - lr:
1.0000e-07
Epoch 99/100
138/138 [==============================] - 60s 438ms/step - loss: 0.1792 -
accuracy: 0.9418 - precision: 0.8750 - recall: 0.8480 - val_loss: 0.3624 -
val_accuracy: 0.8613 - val_precision: 0.9650 - val_recall: 0.6982 - lr:
1.0000e-07
Epoch 100/100
138/138 [==============================] - 61s 441ms/step - loss: 0.1770 -
accuracy: 0.9429 - precision: 0.8708 - recall: 0.8597 - val_loss: 0.3558 -
val_accuracy: 0.8672 - val_precision: 0.9571 - val_recall: 0.7189 - lr:
1.0000e-07
```

[19]:
```python
print("Loss of the model is - " , model.evaluate(x_test,y_test)[0])
print("Accuracy of the model is - " , model.evaluate(x_test,y_test)[1]*100 , "%")
```

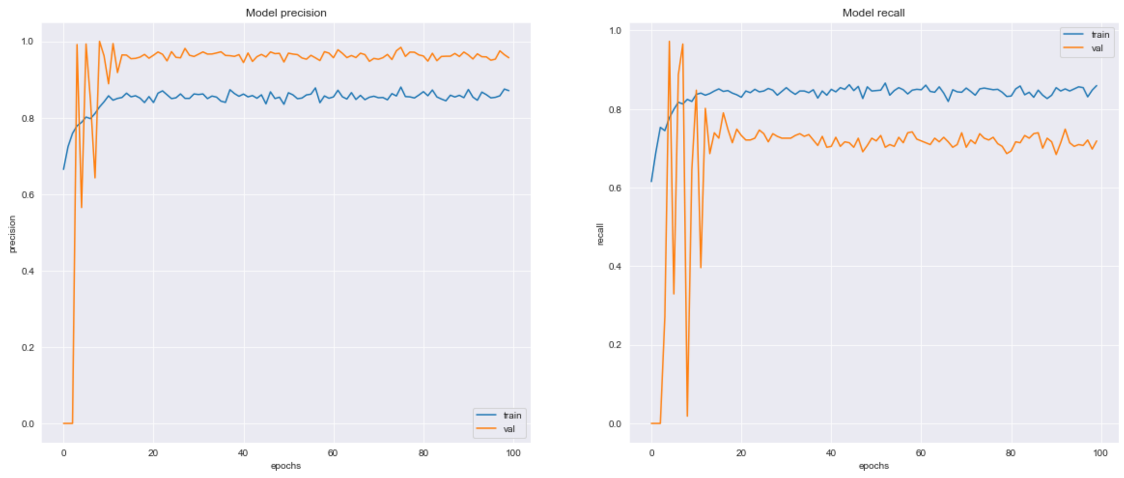
```
13/13 [==============================] - 2s 132ms/step - loss: 0.2212 -
accuracy: 0.9351 - precision: 0.9289 - recall: 0.9423
Loss of the model is -  0.22123439610004425
13/13 [==============================] - 2s 128ms/step - loss: 0.2212 -
accuracy: 0.9351 - precision: 0.9289 - recall: 0.9423
Accuracy of the model is -  93.50961446762085 %
```

0.5.3 Analysis

[20]:
```python
fig, ax = plt.subplots(1, 2, figsize=(20, 8))
ax = ax.ravel()

for i, met in enumerate(['accuracy', 'loss']):
    ax[i].plot(hist.history[met])
    ax[i].plot(hist.history['val_' + met])
    ax[i].set_title('Model {}'.format(met))
    ax[i].set_xlabel('epochs')
    ax[i].set_ylabel(met)
    ax[i].legend(['train', 'val'])
```



[21]:
```python
fig, ax = plt.subplots(1, 2, figsize=(20, 8))
ax = ax.ravel()

for i, met in enumerate(['precision', 'recall']):
    ax[i].plot(hist.history[met])
    ax[i].plot(hist.history['val_' + met])
    ax[i].set_title('Model {}'.format(met))
    ax[i].set_xlabel('epochs')
    ax[i].set_ylabel(met)
    ax[i].legend(['train', 'val'])
```



[22]:
```python
predictions = (model.predict(x_test) > 0.5).astype("int32")
predictions = predictions.reshape(1,-1)[0]
```

```
13/13 [==============================] - 2s 128ms/step
```

```
[23]: print(classification_report(y_test, predictions, target_names = ['Pneumonia␣
       ↪(Class 0)','Normal (Class 1)']))
```

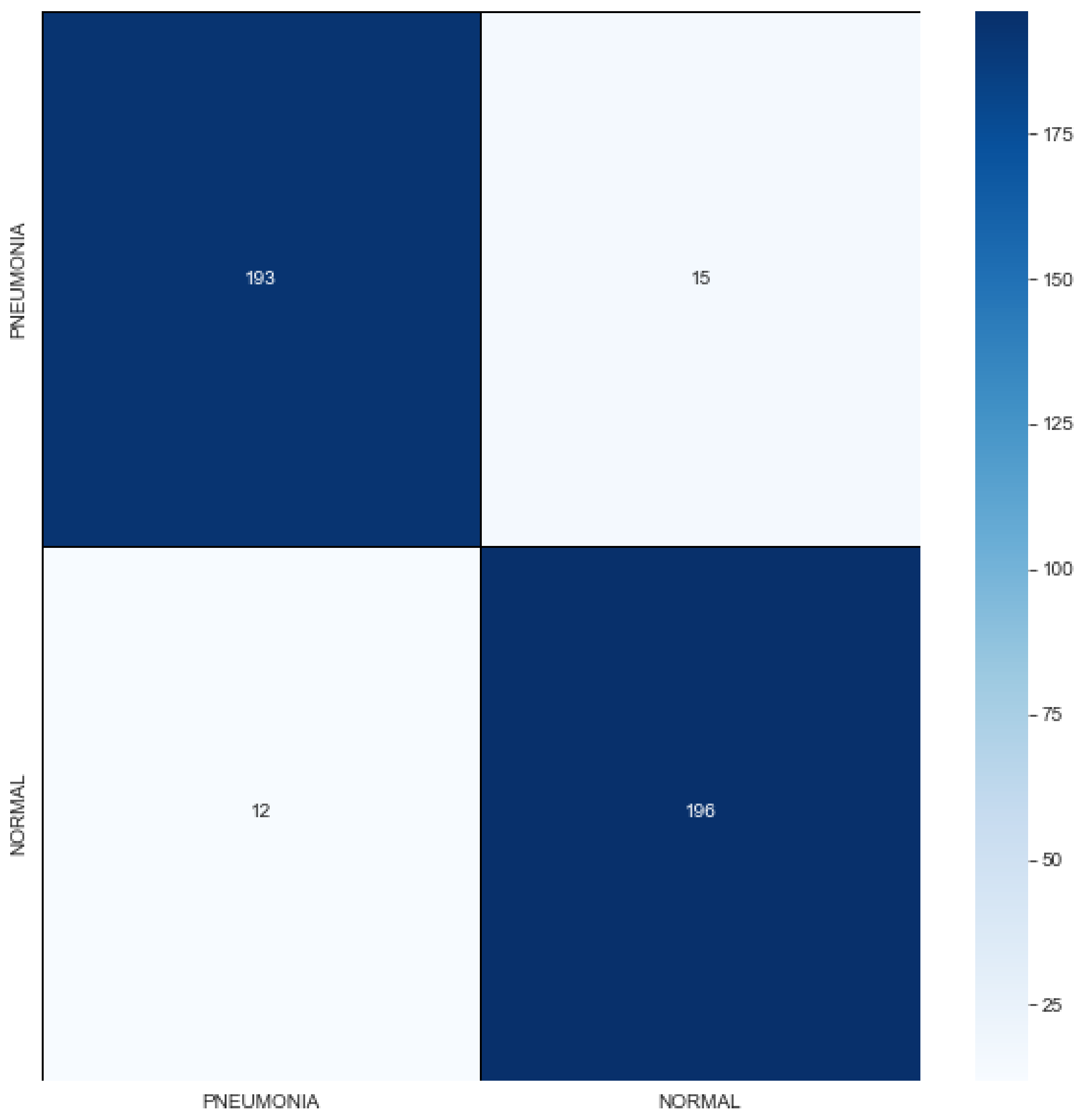
```
                      precision    recall  f1-score   support

Pneumonia (Class 0)       0.94      0.93      0.93       208
   Normal (Class 1)       0.93      0.94      0.94       208

           accuracy                           0.94       416
          macro avg       0.94      0.94      0.94       416
       weighted avg       0.94      0.94      0.94       416
```

```
[24]: cm = confusion_matrix(y_test,predictions)
      cm
```

```
[24]: array([[193,  15],
             [ 12, 196]])
```

```
[25]: cm = pd.DataFrame(cm , index = ['0','1'] , columns = ['0','1'])
      plt.figure(figsize = (10,10))
      sns.heatmap(cm,cmap= "Blues", linecolor = 'black' , linewidth = 1 , annot =␣
       ↪True, fmt='',xticklabels = labels,yticklabels = labels)
```

```
[25]: <AxesSubplot:>
```


PNEUMONIA
NORMAL
193
15
12
196
PNEUMONIA
NORMAL
175
150
125
100
75
50
25