

# Setting Drive

```
from google.colab import drive  
drive.mount('/content/gdrive')
```

## Import Library

```
import numpy as np  
import os  
from glob import glob  
import cv2  
import matplotlib.pyplot as plt  
import seaborn as sns  
from sklearn.metrics import classification_report, confusion_matrix  
from sklearn.model_selection import train_test_split  
  
# TensorFlow  
from keras.callbacks import EarlyStopping  
from tensorflow.keras.utils import to_categorical  
from tensorflow.keras.preprocessing import image  
from tensorflow.keras import models, layers  
from tensorflow.keras.preprocessing.image import ImageDataGenerator  
from keras.models import Sequential  
from tensorflow.keras.layers import Conv2D, Flatten, MaxPool2D, Dense, Dropout  
from tensorflow.keras.optimizers import Adam
```

## Class

```
# Load dataset  
class DatasetLoader:  
    def read_data(data_paths):  
        images = []  
        y = []  
        for data_path in data_paths:  
            labels = ['PNEUMONIA', 'NORMAL']  
            for label in labels:  
                curr_path = data_path + label  
                for img in os.listdir(curr_path):  
                    if ('DS' not in img):  
                        image_path = os.path.join(curr_path, img)  
                        image = cv2.resize(cv2.imread(image_path), (256, 256))  
                        images.append(image)  
                        y.append(label)
```

```

        images.append(image)
        if (label == 'NORMAL'):
            y.append(0)
        else:
            y.append(1)

    images=np.array(images)
    y=to_categorical(y)
    return images,y

# Augmentation and CLAHE
class Preprocessing:
    def augmentation(xtrain, ytrain):
        train_generator = ImageDataGenerator(
            rescale = 1./255,
            shear_range = 0.2,
            zoom_range=0.2,
            horizontal_flip=True,
        ).flow(
            xtrain,
            ytrain,
            batch_size=32
        )

        return train_generator,test_generator

    def enhancement_clahe(data):
        clahe = cv2.createCLAHE(clipLimit = 1, tileGridSize=(8,8))
        images = [];
        for img in data:
            temp = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
            temp = clahe.apply(temp)
            temp = cv2.cvtColor(temp, cv2.COLOR_GRAY2RGB)
            images.append(temp)
        return np.array(images)

# VGG16 with dropout and without dropout
class Vgg16:
    def build():
        model = Sequential()
        model.add(Conv2D(input_shape=(256,256,3),filters=64,kernel_size=(3,3),padding="same",
        model.add(Conv2D(filters=64,kernel_size=(3,3),padding="same", activation="relu"))
        model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
        model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))
        model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))
        model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
        model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
        model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
        model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
        model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

```

```

model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Flatten())
model.add(Dense(units=4096,activation="relu"))
model.add(Dense(units=4096,activation="relu"))
model.add(Dense(units=2, activation="softmax"))

return model

def build_dropout(p):
    model = Sequential()
    model.add(Conv2D(input_shape=(256,256,3),filters=64,kernel_size=(3,3),padding="same",
    model.add(Conv2D(filters=64,kernel_size=(3,3),padding="same", activation="relu"))
    model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
    model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))
    model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))
    model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
    model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
    model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
    model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
    model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
    model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
    model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
    model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
    model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
    model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
    model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
    model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
    model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
    model.add(Flatten())
    model.add(Dense(units=4096,activation="relu"))
    model.add(Dense(units=4096,activation="relu"))
    model.add(Dropout(p))
    model.add(Dense(units=2, activation="softmax"))

return model

class ConfMatrix:
    def heatmap(xtest, ytest):
        y_test_hat = model.predict(xtest, batch_size=4)
        y_test_hat = np.argmax(y_test_hat, axis=1)
        y_test = np.argmax(ytest, axis=1)

        # plot confusion matrix as heatmap

```

```

conf_m = confusion_matrix(y_test, y_test_hat)
plt.figure(figsize=(5,3))
sns.set(font_scale=1.2)
ax = sns.heatmap(conf_m, annot=True, xticklabels=['H', 'P'], yticklabels=['H', 'P'], cbar=False)
plt.yticks(rotation=0)
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
ax.xaxis.set_ticks_position('top')
plt.title('Confusion matrix - test data\n(H - healthy/normal, P - pneumonia)')
plt.show()
return plt

def classification_report(ytest):
    y_test_hat = model.predict(xtest, batch_size=4)
    y_test_hat = np.argmax(y_test_hat, axis=1)
    y_test = np.argmax(ytest, axis=1)

    clas_r = classification_report(y_test, y_test_hat)
    return clas_r

```

## Load Data

```

path = 'gdrive/MyDrive/Dataset/Pneumonia Full/chest_xray/'

# train directory
train_folder=path+"train/"
train_normal_dir=train_folder+"NORMAL/"
train_pneu_dir=train_folder+"PNEUMONIA/"

# test directory
test_folder=path+"test/"
test_normal_dir=test_folder+"NORMAL/"
test_pneu_dir=test_folder+"PNEUMONIA/"

# validation directory
val_folder=path+"val/"
val_normal_dir=val_folder+"NORMAL/"
val_pneu_dir=val_folder+"PNEUMONIA/"

all_folder=[train_folder,test_folder,val_folder]

# img size
img_width=256
img_height=256

print("[INFO] loading...")
x, y = DatasetLoader.read_data(all_folder)

```

```

print("[INFO] Load image done")

[INFO] loading...
[INFO] Load image done

xtrain, xtest, ytrain, ytest = train_test_split(x,y,test_size=0.1,random_state = 42,shuffle = True)

print("X_train shape: {}".format(xtrain.shape))
print("X_test shape: {}".format(xtest.shape))
print("y_train shape: {}".format(ytrain.shape))
print("y_test shape: {}".format(ytest.shape))

X_train shape: (5270, 256, 256, 3)
X_test shape: (586, 256, 256, 3)
y_train shape: (5270, 2)
y_test shape: (586, 2)

```

## Model Spec

```

model = Vgg16.build()
model.summary()

```

Model: "sequential"

Layer (type)	Output Shape	Param #
<hr/>		
conv2d (Conv2D)	(None, 256, 256, 64)	1792
conv2d_1 (Conv2D)	(None, 256, 256, 64)	36928
max_pooling2d (MaxPooling2D )	(None, 128, 128, 64)	0
conv2d_2 (Conv2D)	(None, 128, 128, 128)	73856
conv2d_3 (Conv2D)	(None, 128, 128, 128)	147584
max_pooling2d_1 (MaxPooling 2D)	(None, 64, 64, 128)	0
conv2d_4 (Conv2D)	(None, 64, 64, 256)	295168
conv2d_5 (Conv2D)	(None, 64, 64, 256)	590080
conv2d_6 (Conv2D)	(None, 64, 64, 256)	590080
max_pooling2d_2 (MaxPooling 2D)	(None, 32, 32, 256)	0
conv2d_7 (Conv2D)	(None, 32, 32, 512)	1180160

conv2d_8 (Conv2D)	(None, 32, 32, 512)	2359808
conv2d_9 (Conv2D)	(None, 32, 32, 512)	2359808
max_pooling2d_3 (MaxPooling 2D)	(None, 16, 16, 512)	0
conv2d_10 (Conv2D)	(None, 16, 16, 512)	2359808
conv2d_11 (Conv2D)	(None, 16, 16, 512)	2359808
conv2d_12 (Conv2D)	(None, 16, 16, 512)	2359808
max_pooling2d_4 (MaxPooling 2D)	(None, 8, 8, 512)	0
flatten (Flatten)	(None, 32768)	0
dense (Dense)	(None, 4096)	134221824
dense_1 (Dense)	(None, 4096)	16781312
dense_2 (Dense)	(None, 2)	8194

```
=====
Total params: 165,726,018
Trainable params: 165,726,018
Non-trainable params: 0
```

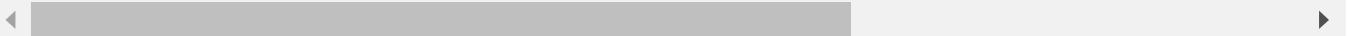
## Train

## Normal

Batch size 5 + Lr 0.0001 + Epoch 10

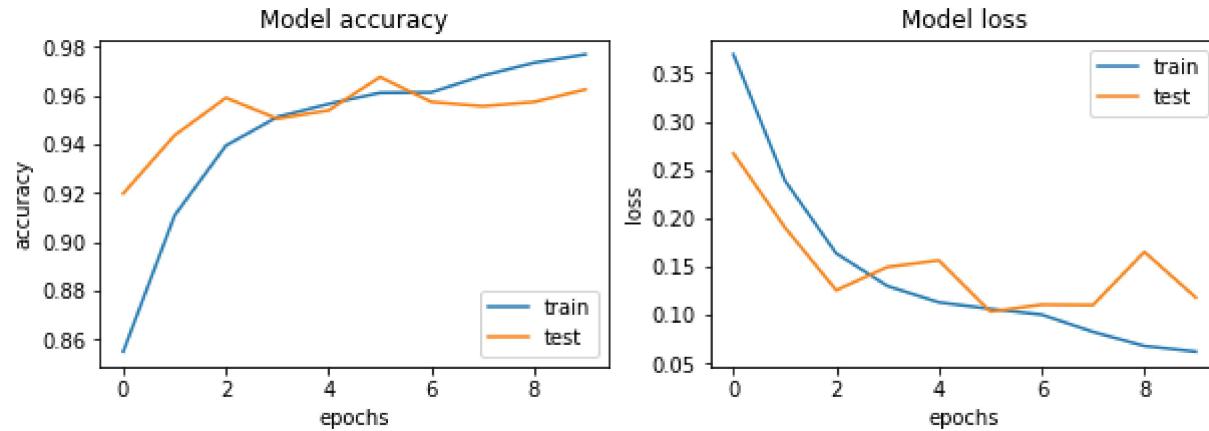
```
model = Vgg16.build()
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.0001)
history = model.fit(xtrain,ytrain,epochs=10, batch_size=5, verbose=True,validation_
Epoch 1/10
1054/1054 [=====] - 427s 392ms/step - loss: 0.3693 - accuracy: 0.2389
Epoch 2/10
1054/1054 [=====] - 414s 393ms/step - loss: 0.2389 - accuracy: 0.1635
Epoch 3/10
1054/1054 [=====] - 413s 392ms/step - loss: 0.1635 - accuracy: 0.1297
Epoch 4/10
1054/1054 [=====] - 417s 396ms/step - loss: 0.1297 - accuracy: 0.1297
Epoch 5/10
```

```
1054/1054 [=====] - 412s 391ms/step - loss: 0.1128 - accuracy: Epoch 6/10
1054/1054 [=====] - 413s 392ms/step - loss: 0.1061 - accuracy: Epoch 7/10
1054/1054 [=====] - 412s 391ms/step - loss: 0.1001 - accuracy: Epoch 8/10
1054/1054 [=====] - 417s 396ms/step - loss: 0.0823 - accuracy: Epoch 9/10
1054/1054 [=====] - 412s 391ms/step - loss: 0.0678 - accuracy: Epoch 10/10
1054/1054 [=====] - 412s 391ms/step - loss: 0.0619 - accuracy:
```



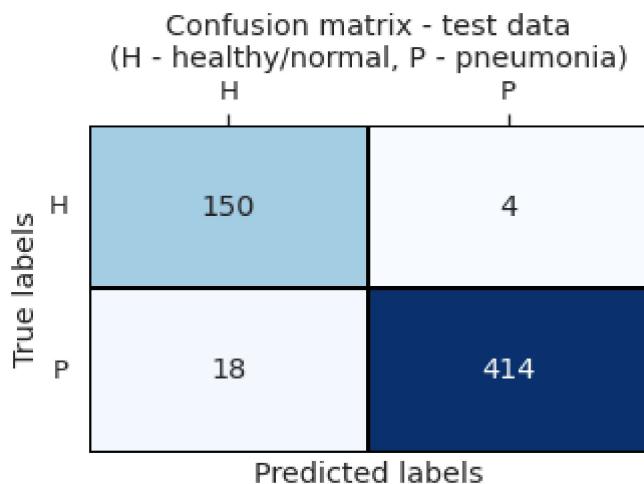
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision      recall    f1-score   support
```

## Batch size 5 + Lr 0.0001 + Epoch 10 + Probability 0.2

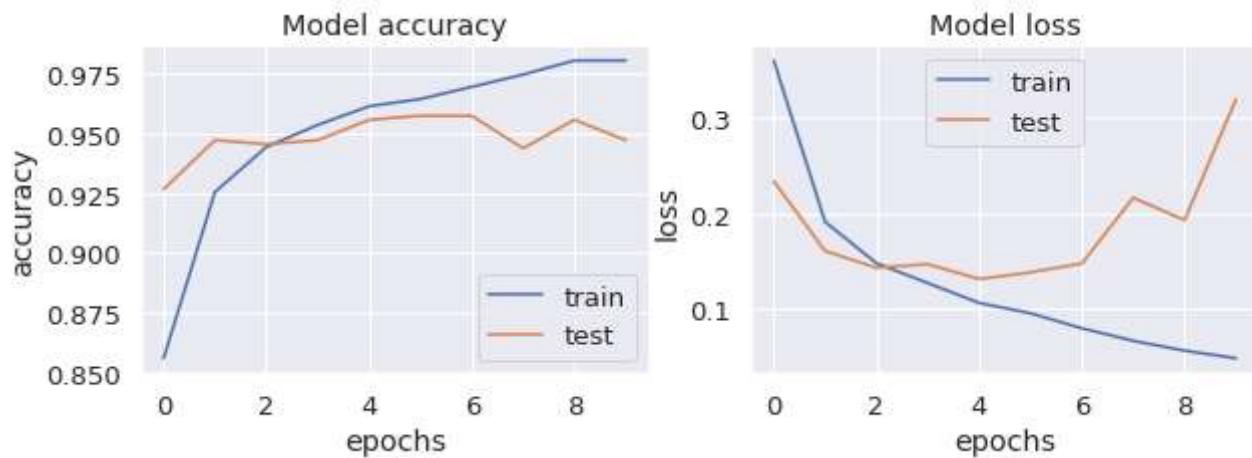
```
accuracy           0.96      586
model = Vgg16.build_dropout(0.2)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.0001),metrics
history = model.fit(xtrain,ytrain,epochs=10, batch_size=5, verbose=True,validation_data=(xte
Epoch 1/10
1054/1054 [=====] - 417s 394ms/step - loss: 0.3601 - accuracy:
Epoch 2/10
1054/1054 [=====] - 414s 392ms/step - loss: 0.1910 - accuracy:
Epoch 3/10
1054/1054 [=====] - 413s 392ms/step - loss: 0.1476 - accuracy:
Epoch 4/10
1054/1054 [=====] - 412s 391ms/step - loss: 0.1268 - accuracy:
Epoch 5/10
1054/1054 [=====] - 410s 389ms/step - loss: 0.1060 - accuracy:
Epoch 6/10
1054/1054 [=====] - 409s 389ms/step - loss: 0.0952 - accuracy:
Epoch 7/10
1054/1054 [=====] - 410s 389ms/step - loss: 0.0794 - accuracy:
Epoch 8/10
1054/1054 [=====] - 410s 389ms/step - loss: 0.0663 - accuracy:
Epoch 9/10
1054/1054 [=====] - 410s 389ms/step - loss: 0.0560 - accuracy:
Epoch 10/10
1054/1054 [=====] - 409s 388ms/step - loss: 0.0480 - accuracy:
```



```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

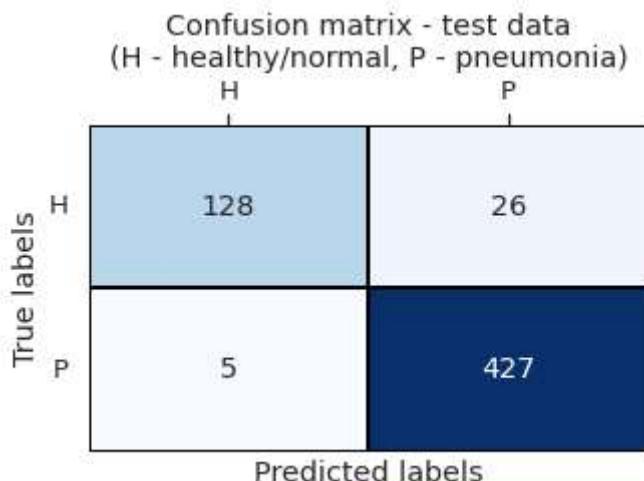
for i, met in enumerate(['accuracy', 'loss']):
    ax[i].plot(history.history[met])
```

```
ax[i].plot(history.history['val_' + met])
ax[i].set_title('Model {}'.format(met))
ax[i].set_xlabel('epochs')
ax[i].set_ylabel(met)
ax[i].legend(['train', 'test'])
```



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyplot.py'>
```

	precision	recall	f1-score	support
0	0.96	0.83	0.89	154
1	0.94	0.99	0.96	432
accuracy			0.95	586
macro avg	0.95	0.91	0.93	586
weighted avg	0.95	0.95	0.95	586

## Batch size 5 + Lr 0.0001 + Epoch 10 + Probability 0.5

```
model = Vgg16.build_dropout(0.5)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.0001),metrics
history = model.fit(xtrain,ytrain,epochs=10, batch_size=5, verbose=True,validation_data=(xtes

Epoch 1/10
1054/1054 [=====] - 173s 151ms/step - loss: 0.3199 - accuracy:
Epoch 2/10
1054/1054 [=====] - 161s 153ms/step - loss: 0.1621 - accuracy:
Epoch 3/10
1054/1054 [=====] - 160s 152ms/step - loss: 0.1456 - accuracy:
Epoch 4/10
1054/1054 [=====] - 160s 151ms/step - loss: 0.1168 - accuracy:
Epoch 5/10
1054/1054 [=====] - 160s 152ms/step - loss: 0.1038 - accuracy:
Epoch 6/10
1054/1054 [=====] - 159s 151ms/step - loss: 0.0919 - accuracy:
Epoch 7/10
1054/1054 [=====] - 159s 151ms/step - loss: 0.0818 - accuracy:
Epoch 8/10
1054/1054 [=====] - 158s 150ms/step - loss: 0.0699 - accuracy:
Epoch 9/10
1054/1054 [=====] - 160s 151ms/step - loss: 0.0568 - accuracy:
Epoch 10/10
1054/1054 [=====] - 159s 151ms/step - loss: 0.0569 - accuracy:
```



```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

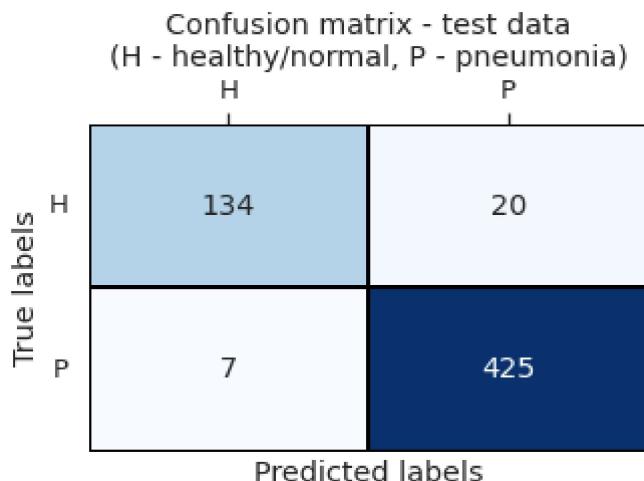
for i, met in enumerate(['accuracy', 'loss']):
    ax[i].plot(history.history[met])
    ax[i].plot(history.history['val_' + met])
    ax[i].set_title('Model {}'.format(met))
    ax[i].set_xlabel('epochs')
    ax[i].set_ylabel(met)
    ax[i].legend(['train', 'test'])
```

## Model accuracy

## Model loss

```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)

report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyplot.py'>
precision    recall    f1-score    support
          0       0.95      0.87      0.91      154
          1       0.96      0.98      0.97      432
   accuracy                           0.95      586
  macro avg       0.95      0.93      0.94      586
weighted avg       0.95      0.95      0.95      586
```

## Batch size 5 + Lr 0.0001 + Epoch 20

```
model = Vgg16.build()
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.0001),metrics
history = model.fit(xtrain,ytrain,epochs=20, batch_size=5, verbose=True,validation_data=(xtes

Epoch 1/20
1054/1054 [=====] - 164s 155ms/step - loss: 0.3583 - accuracy: 0.1924
Epoch 2/20
1054/1054 [=====] - 162s 154ms/step - loss: 0.1924 - accuracy: 0.1433
Epoch 3/20
1054/1054 [=====] - 161s 152ms/step - loss: 0.1433 - accuracy: 0.1252
Epoch 4/20
1054/1054 [=====] - 161s 152ms/step - loss: 0.1252 - accuracy: 0.1100
Epoch 5/20
1054/1054 [=====] - 160s 152ms/step - loss: 0.1100 - accuracy: 0.1000
Epoch 6/20
```

```
1054/1054 [=====] - 160s 152ms/step - loss: 0.0953 - accuracy: Epoch 7/20
1054/1054 [=====] - 160s 152ms/step - loss: 0.1021 - accuracy: Epoch 8/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0742 - accuracy: Epoch 9/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0689 - accuracy: Epoch 10/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0591 - accuracy: Epoch 11/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0491 - accuracy: Epoch 12/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0389 - accuracy: Epoch 13/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0403 - accuracy: Epoch 14/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0361 - accuracy: Epoch 15/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0286 - accuracy: Epoch 16/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0310 - accuracy: Epoch 17/20
1054/1054 [=====] - 159s 150ms/step - loss: 0.0180 - accuracy: Epoch 18/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0235 - accuracy: Epoch 19/20
1054/1054 [=====] - 158s 150ms/step - loss: 0.0244 - accuracy: Epoch 20/20
1054/1054 [=====] - 158s 150ms/step - loss: 0.0172 - accuracy:
```



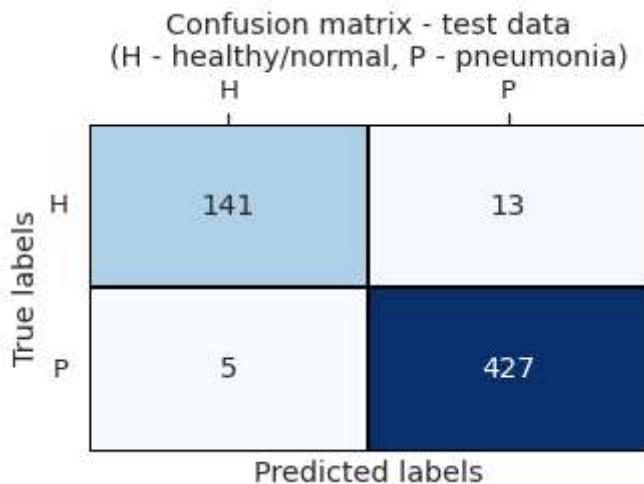
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision    recall    f1-score    support
          0       0.97      0.92      0.94      154
          1       0.97      0.99      0.98      432
   accuracy                           0.97      586
  macro avg       0.97      0.95      0.96      586
weighted avg       0.97      0.97      0.97      586
```

Batch size 5 + Lr 0.0001 + Epoch 20 + Probability 0.2

```
model = Vgg16.build_dropout(0.2)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.0001),metrics
history = model.fit(xtrain,ytrain,epochs=20, batch_size=5, verbose=True,validation_data=(xtes
```

```
Epoch 1/20
1054/1054 [=====] - 163s 154ms/step - loss: 0.4128 - accuracy: 0.2156
Epoch 2/20
1054/1054 [=====] - 166s 157ms/step - loss: 0.2156 - accuracy: 0.4128
Epoch 3/20
1054/1054 [=====] - 160s 152ms/step - loss: 0.1502 - accuracy: 0.586
Epoch 4/20
1054/1054 [=====] - 160s 152ms/step - loss: 0.1277 - accuracy: 0.687
Epoch 5/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.1065 - accuracy: 0.759
```

```
Epoch 6/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0942 - accuracy: 0.8054
Epoch 7/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0821 - accuracy: 0.8154
Epoch 8/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0662 - accuracy: 0.8354
Epoch 9/20
1054/1054 [=====] - 158s 150ms/step - loss: 0.0654 - accuracy: 0.8354
Epoch 10/20
1054/1054 [=====] - 158s 150ms/step - loss: 0.0480 - accuracy: 0.8654
Epoch 11/20
1054/1054 [=====] - 158s 150ms/step - loss: 0.0595 - accuracy: 0.8454
Epoch 12/20
1054/1054 [=====] - 158s 150ms/step - loss: 0.0353 - accuracy: 0.8854
Epoch 13/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0340 - accuracy: 0.8954
Epoch 14/20
1054/1054 [=====] - 158s 150ms/step - loss: 0.0311 - accuracy: 0.8954
Epoch 15/20
1054/1054 [=====] - 158s 150ms/step - loss: 0.0237 - accuracy: 0.9054
Epoch 16/20
1054/1054 [=====] - 158s 150ms/step - loss: 0.0180 - accuracy: 0.9154
Epoch 17/20
1054/1054 [=====] - 159s 151ms/step - loss: 0.0281 - accuracy: 0.9154
Epoch 18/20
1054/1054 [=====] - 158s 150ms/step - loss: 0.0219 - accuracy: 0.9154
Epoch 19/20
1054/1054 [=====] - 163s 155ms/step - loss: 0.0260 - accuracy: 0.9154
Epoch 20/20
1054/1054 [=====] - 158s 150ms/step - loss: 0.0187 - accuracy: 0.9254
```



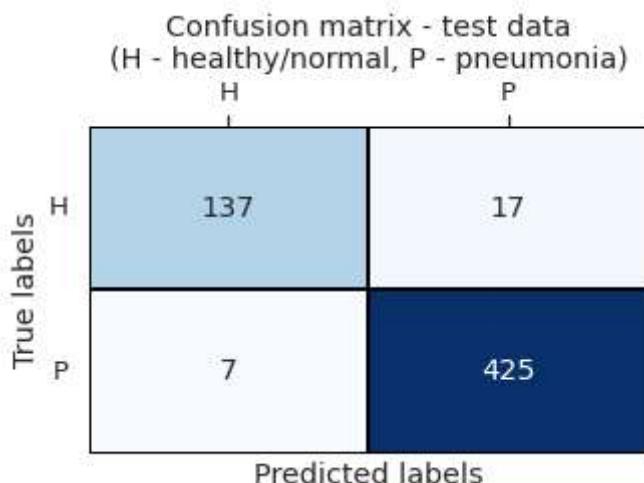
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```

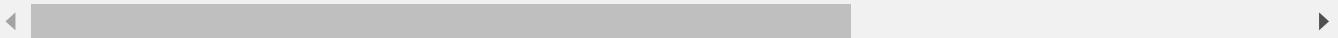


```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision    recall    f1-score    support
0            0.95      0.89      0.92      154
1            0.96      0.98      0.97      432
accuracy                           0.96      586
macro avg       0.96      0.94      0.95      586
weighted avg    0.96      0.96      0.96      586
```

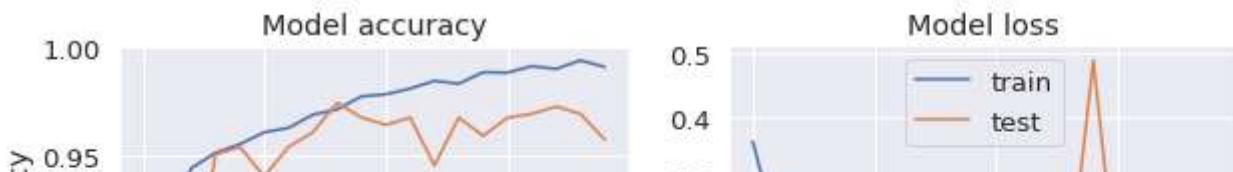
Batch size 5 + Lr 0.0001 + Epoch 20 + Probability 0.5

```
model = Vgg16.build_dropout(0.5)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.0001))
history = model.fit(xtrain,ytrain,epochs=20, batch_size=5, verbose=True, validation_
Epoch 1/20
1054/1054 [=====] - 163s 154ms/step - loss: 0.3599 - accuracy: 0.645
Epoch 2/20
1054/1054 [=====] - 160s 152ms/step - loss: 0.2132 - accuracy: 0.755
Epoch 3/20
1054/1054 [=====] - 160s 152ms/step - loss: 0.1539 - accuracy: 0.815
Epoch 4/20
1054/1054 [=====] - 160s 151ms/step - loss: 0.1288 - accuracy: 0.855
Epoch 5/20
```

```
1054/1054 [=====] - 159s 151ms/step - loss: 0.1204 - accuracy:  
Epoch 6/20  
1054/1054 [=====] - 159s 151ms/step - loss: 0.1093 - accuracy:  
Epoch 7/20  
1054/1054 [=====] - 159s 151ms/step - loss: 0.0942 - accuracy:  
Epoch 8/20  
1054/1054 [=====] - 158s 150ms/step - loss: 0.0806 - accuracy:  
Epoch 9/20  
1054/1054 [=====] - 158s 150ms/step - loss: 0.0804 - accuracy:  
Epoch 10/20  
1054/1054 [=====] - 158s 150ms/step - loss: 0.0655 - accuracy:  
Epoch 11/20  
1054/1054 [=====] - 158s 150ms/step - loss: 0.0600 - accuracy:  
Epoch 12/20  
1054/1054 [=====] - 158s 150ms/step - loss: 0.0492 - accuracy:  
Epoch 13/20  
1054/1054 [=====] - 158s 150ms/step - loss: 0.0413 - accuracy:  
Epoch 14/20  
1054/1054 [=====] - 158s 150ms/step - loss: 0.0472 - accuracy:  
Epoch 15/20  
1054/1054 [=====] - 158s 150ms/step - loss: 0.0316 - accuracy:  
Epoch 16/20  
1054/1054 [=====] - 158s 150ms/step - loss: 0.0376 - accuracy:  
Epoch 17/20  
1054/1054 [=====] - 158s 150ms/step - loss: 0.0219 - accuracy:  
Epoch 18/20  
1054/1054 [=====] - 158s 150ms/step - loss: 0.0316 - accuracy:  
Epoch 19/20  
1054/1054 [=====] - 157s 149ms/step - loss: 0.0163 - accuracy:  
Epoch 20/20  
1054/1054 [=====] - 158s 150ms/step - loss: 0.0242 - accuracy:
```

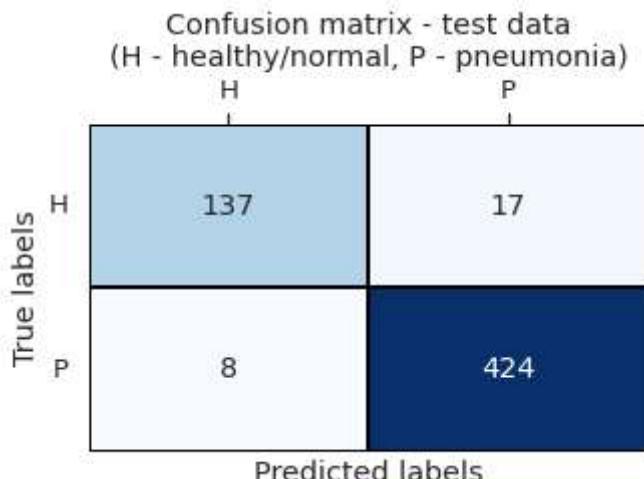


```
# Visualize Loss and Accuracy Rates  
fig, ax = plt.subplots(1, 2, figsize=(10, 3))  
ax = ax.ravel()  
  
for i, met in enumerate(['accuracy', 'loss']):  
    ax[i].plot(history.history[met])  
    ax[i].plot(history.history['val_' + met])  
    ax[i].set_title('Model {}'.format(met))  
    ax[i].set_xlabel('epochs')  
    ax[i].set_ylabel(met)  
    ax[i].legend(['train', 'test'])
```



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)

report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision    recall    f1-score    support
0            0.94     0.89     0.92      154
1            0.96     0.98     0.97      432
accuracy                           0.96      586
macro avg       0.95     0.94     0.94      586
weighted avg    0.96     0.96     0.96      586
```

## Batch size 5 + Lr 0.00001 + Epoch 10

```
model = Vgg16.build()
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.00001),metric
history = model.fit(xtrain,ytrain,epochs=10, batch_size=5, verbose=True,validation_data=(xte
Epoch 1/10
1054/1054 [=====] - 419s 384ms/step - loss: 0.2303 - accuracy:
Epoch 2/10
1054/1054 [=====] - 400s 379ms/step - loss: 0.1309 - accuracy:
Epoch 3/10
1054/1054 [=====] - 402s 381ms/step - loss: 0.1079 - accuracy:
Epoch 4/10
1054/1054 [=====] - 402s 382ms/step - loss: 0.0935 - accuracy:
```

```

Epoch 5/10
1054/1054 [=====] - 402s 382ms/step - loss: 0.0819 - accuracy:
Epoch 6/10
1054/1054 [=====] - 403s 382ms/step - loss: 0.0677 - accuracy:
Epoch 7/10
1054/1054 [=====] - 402s 382ms/step - loss: 0.0580 - accuracy:
Epoch 8/10
1054/1054 [=====] - 407s 386ms/step - loss: 0.0550 - accuracy:
Epoch 9/10
1054/1054 [=====] - 402s 381ms/step - loss: 0.0383 - accuracy:
Epoch 10/10
1054/1054 [=====] - 398s 378ms/step - loss: 0.0379 - accuracy:

```

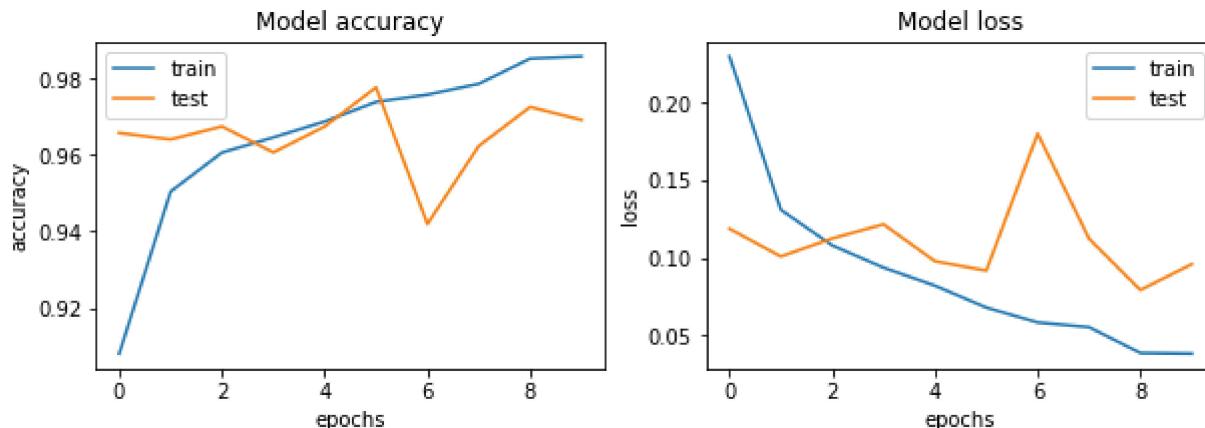


```

# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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

```



```

heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)

```

```

report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)

```

		Confusion matrix - test data	
		(H - healthy/normal, P - pneumonia)	
		H	P
True labels	H	143	11
	P	7	425
Predicted labels			

```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision      recall    f1-score   support
0            0.95      0.93      0.94     154
1            0.97      0.98      0.98     432
```

## Batch size 5 + Lr 0.00001 + Epoch 10 + Probability 0.2

```
weighted avg      0.97      0.97      0.97     586
```

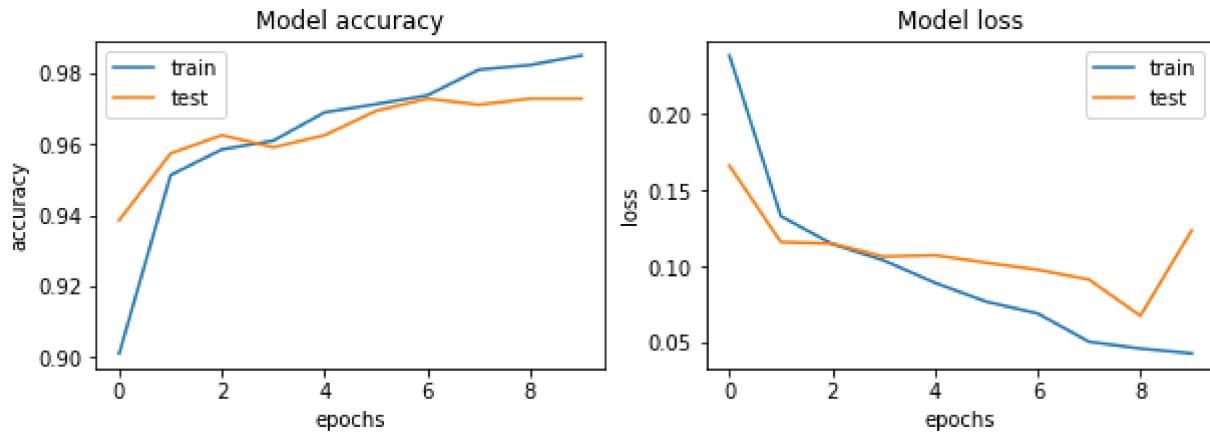
```
model = Vgg16.build_dropout(0.2)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.00001),metric
history = model.fit(xtrain,ytrain,epochs=10, batch_size=5, verbose=True,validation_data=(xtes

Epoch 1/10
1054/1054 [=====] - 175s 154ms/step - loss: 0.2385 - accuracy:
Epoch 2/10
1054/1054 [=====] - 173s 164ms/step - loss: 0.1329 - accuracy:
Epoch 3/10
1054/1054 [=====] - 169s 160ms/step - loss: 0.1147 - accuracy:
Epoch 4/10
1054/1054 [=====] - 168s 160ms/step - loss: 0.1040 - accuracy:
Epoch 5/10
1054/1054 [=====] - 169s 160ms/step - loss: 0.0892 - accuracy:
Epoch 6/10
1054/1054 [=====] - 169s 160ms/step - loss: 0.0768 - accuracy:
Epoch 7/10
1054/1054 [=====] - 168s 160ms/step - loss: 0.0690 - accuracy:
Epoch 8/10
1054/1054 [=====] - 169s 160ms/step - loss: 0.0505 - accuracy:
Epoch 9/10
1054/1054 [=====] - 168s 160ms/step - loss: 0.0460 - accuracy:
Epoch 10/10
1054/1054 [=====] - 169s 160ms/step - loss: 0.0428 - accuracy:
```



```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()
```

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)  
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)  
print(report_normal)
```

## Batch size 5 + Lr 0.00001 + Epoch 10 + Probability 0.5

```
model = Vgg16.build_dropout(0.5)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.00001),metric
history = model.fit(xtrain,ytrain,epochs=10, batch_size=5, verbose=True,validation_data=(xte

Epoch 1/10
1054/1054 [=====] - 170s 160ms/step - loss: 0.2552 - accuracy:
Epoch 2/10
1054/1054 [=====] - 170s 161ms/step - loss: 0.1417 - accuracy:
Epoch 3/10
1054/1054 [=====] - 168s 160ms/step - loss: 0.1275 - accuracy:
Epoch 4/10
1054/1054 [=====] - 170s 161ms/step - loss: 0.1022 - accuracy:
Epoch 5/10
1054/1054 [=====] - 170s 161ms/step - loss: 0.0944 - accuracy:
Epoch 6/10
1054/1054 [=====] - 170s 161ms/step - loss: 0.0802 - accuracy:
Epoch 7/10
1054/1054 [=====] - 170s 161ms/step - loss: 0.0680 - accuracy:
Epoch 8/10
1054/1054 [=====] - 169s 161ms/step - loss: 0.0566 - accuracy:
Epoch 9/10
1054/1054 [=====] - 170s 161ms/step - loss: 0.0542 - accuracy:
Epoch 10/10
1054/1054 [=====] - 169s 160ms/step - loss: 0.0437 - accuracy:
```

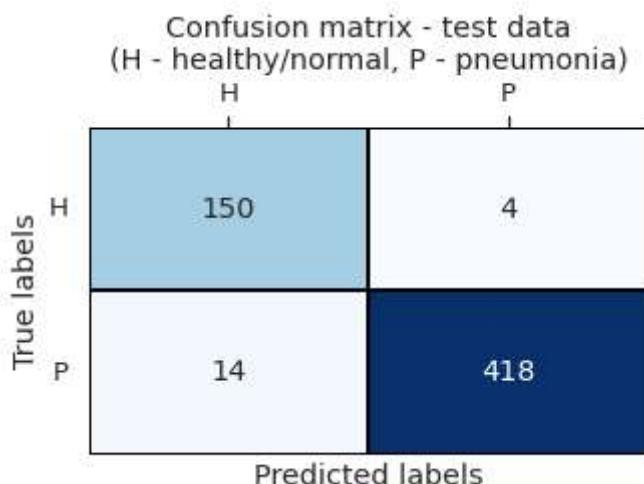
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



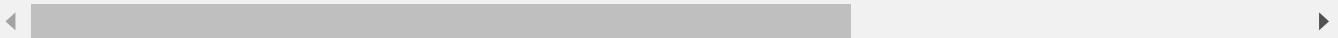
```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision    recall    f1-score    support
0            0.91      0.97      0.94      154
1            0.99      0.97      0.98      432
accuracy                           0.97      586
macro avg       0.95      0.97      0.96      586
weighted avg    0.97      0.97      0.97      586
```

## Batch size 5 + Lr 0.00001 + Epoch 20

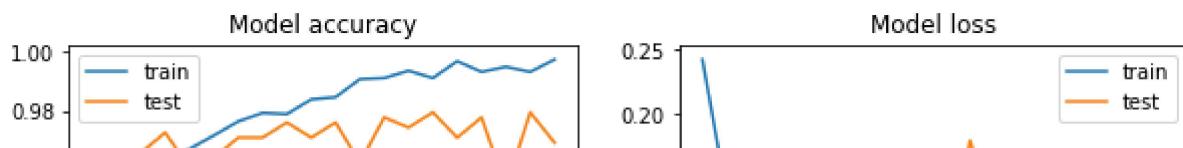
```
model = Vgg16.build()
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.00001),metric
history = model.fit(xtrain,ytrain,epochs=20, batch_size=5, verbose=True,validation_data=(xtes

Epoch 1/20
1054/1054 [=====] - 436s 400ms/step - loss: 0.2425 - accuracy:
Epoch 2/20
1054/1054 [=====] - 421s 399ms/step - loss: 0.1356 - accuracy:
Epoch 3/20
1054/1054 [=====] - 420s 398ms/step - loss: 0.1170 - accuracy:
Epoch 4/20
1054/1054 [=====] - 420s 398ms/step - loss: 0.0981 - accuracy:
Epoch 5/20
```

```
1054/1054 [=====] - 419s 398ms/step - loss: 0.0907 - accuracy:  
Epoch 6/20  
1054/1054 [=====] - 423s 401ms/step - loss: 0.0736 - accuracy:  
Epoch 7/20  
1054/1054 [=====] - 419s 398ms/step - loss: 0.0594 - accuracy:  
Epoch 8/20  
1054/1054 [=====] - 419s 397ms/step - loss: 0.0581 - accuracy:  
Epoch 9/20  
1054/1054 [=====] - 418s 397ms/step - loss: 0.0565 - accuracy:  
Epoch 10/20  
1054/1054 [=====] - 418s 397ms/step - loss: 0.0380 - accuracy:  
Epoch 11/20  
1054/1054 [=====] - 419s 397ms/step - loss: 0.0357 - accuracy:  
Epoch 12/20  
1054/1054 [=====] - 418s 397ms/step - loss: 0.0231 - accuracy:  
Epoch 13/20  
1054/1054 [=====] - 418s 397ms/step - loss: 0.0246 - accuracy:  
Epoch 14/20  
1054/1054 [=====] - 419s 397ms/step - loss: 0.0185 - accuracy:  
Epoch 15/20  
1054/1054 [=====] - 420s 399ms/step - loss: 0.0227 - accuracy:  
Epoch 16/20  
1054/1054 [=====] - 420s 399ms/step - loss: 0.0102 - accuracy:  
Epoch 17/20  
1054/1054 [=====] - 420s 398ms/step - loss: 0.0186 - accuracy:  
Epoch 18/20  
1054/1054 [=====] - 420s 399ms/step - loss: 0.0157 - accuracy:  
Epoch 19/20  
1054/1054 [=====] - 419s 398ms/step - loss: 0.0189 - accuracy:  
Epoch 20/20  
1054/1054 [=====] - 423s 401ms/step - loss: 0.0096 - accuracy:
```

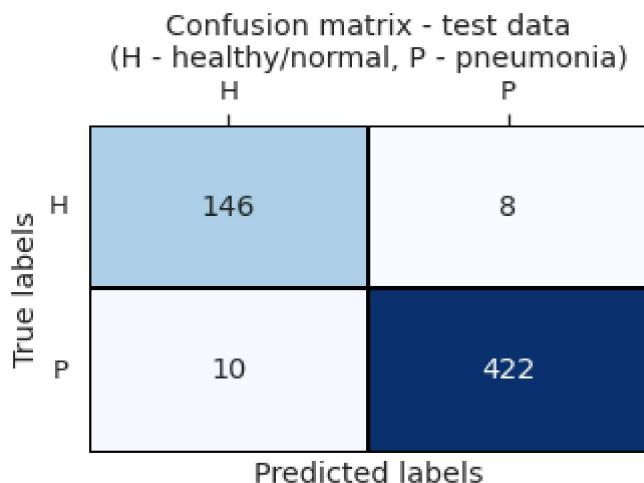


```
# Visualize Loss and Accuracy Rates  
fig, ax = plt.subplots(1, 2, figsize=(10, 3))  
ax = ax.ravel()  
  
for i, met in enumerate(['accuracy', 'loss']):  
    ax[i].plot(history.history[met])  
    ax[i].plot(history.history['val_' + met])  
    ax[i].set_title('Model {}'.format(met))  
    ax[i].set_xlabel('epochs')  
    ax[i].set_ylabel(met)  
    ax[i].legend(['train', 'test'])
```



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision      recall    f1-score   support
0            0.94      0.95      0.94      154
1            0.98      0.98      0.98      432
accuracy                           0.97      586
macro avg       0.96      0.96      0.96      586
weighted avg    0.97      0.97      0.97      586
```

Batch size 5 + Lr 0.00001 + Epoch 20 + Probability 0.2

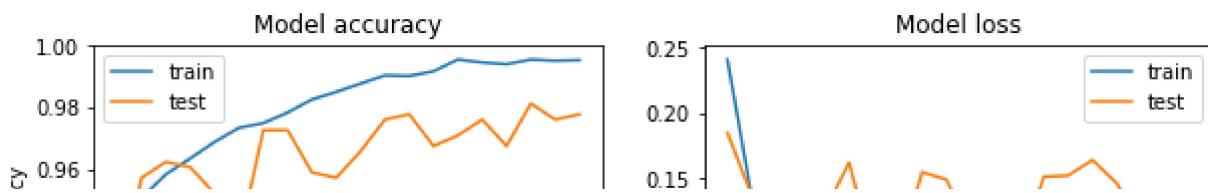
```
model = Vgg16.build_dropout(0.2)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.00001),metric
history = model.fit(xtrain,ytrain,epochs=20, batch_size=5, verbose=True,validation_data=(xte
Epoch 1/20
1054/1054 [=====] - 417s 382ms/step - loss: 0.2413 - accuracy:
Epoch 2/20
1054/1054 [=====] - 398s 378ms/step - loss: 0.1353 - accuracy:
Epoch 3/20
1054/1054 [=====] - 398s 378ms/step - loss: 0.1113 - accuracy:
Epoch 4/20
1054/1054 [=====] - 398s 378ms/step - loss: 0.0996 - accuracy:
```

```
Epoch 5/20
1054/1054 [=====] - 398s 377ms/step - loss: 0.0865 - accuracy: 0.9135
Epoch 6/20
1054/1054 [=====] - 397s 377ms/step - loss: 0.0771 - accuracy: 0.9135
Epoch 7/20
1054/1054 [=====] - 398s 378ms/step - loss: 0.0664 - accuracy: 0.9135
Epoch 8/20
1054/1054 [=====] - 398s 378ms/step - loss: 0.0580 - accuracy: 0.9135
Epoch 9/20
1054/1054 [=====] - 398s 377ms/step - loss: 0.0469 - accuracy: 0.9135
Epoch 10/20
1054/1054 [=====] - 397s 377ms/step - loss: 0.0387 - accuracy: 0.9135
Epoch 11/20
1054/1054 [=====] - 398s 377ms/step - loss: 0.0358 - accuracy: 0.9135
Epoch 12/20
1054/1054 [=====] - 398s 378ms/step - loss: 0.0267 - accuracy: 0.9135
Epoch 13/20
1054/1054 [=====] - 397s 377ms/step - loss: 0.0257 - accuracy: 0.9135
Epoch 14/20
1054/1054 [=====] - 397s 377ms/step - loss: 0.0230 - accuracy: 0.9135
Epoch 15/20
1054/1054 [=====] - 398s 377ms/step - loss: 0.0131 - accuracy: 0.9135
Epoch 16/20
1054/1054 [=====] - 398s 377ms/step - loss: 0.0159 - accuracy: 0.9135
Epoch 17/20
1054/1054 [=====] - 398s 378ms/step - loss: 0.0155 - accuracy: 0.9135
Epoch 18/20
1054/1054 [=====] - 397s 377ms/step - loss: 0.0130 - accuracy: 0.9135
Epoch 19/20
1054/1054 [=====] - 397s 377ms/step - loss: 0.0154 - accuracy: 0.9135
Epoch 20/20
1054/1054 [=====] - 397s 377ms/step - loss: 0.0155 - accuracy: 0.9135
```



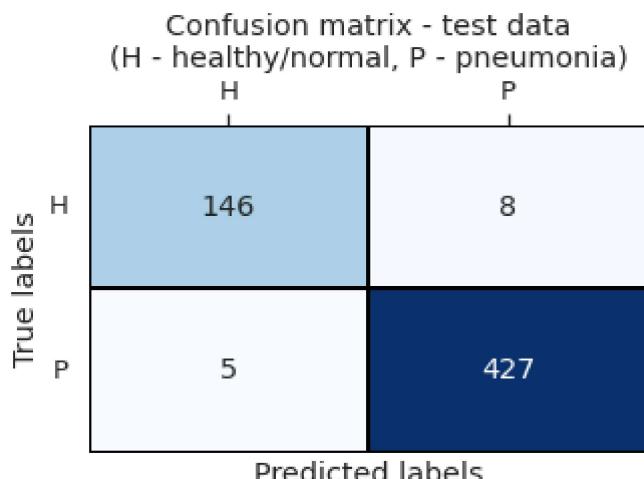
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision      recall    f1-score    support
          0       0.97      0.95      0.96     154
          1       0.98      0.99      0.99     432
   accuracy                           0.98     586
  macro avg       0.97      0.97      0.97     586
weighted avg       0.98      0.98      0.98     586
```

Batch size 5 + Lr 0.00001 + Epoch 20 + Probability 0.5

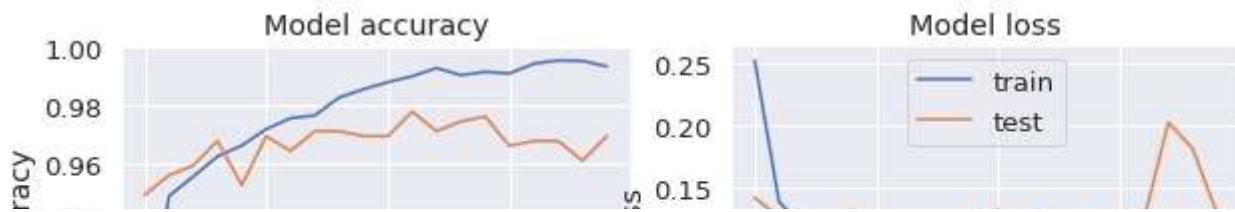
```
model = Vgg16.build_dropout(0.5)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.00001),metric
history = model.fit(xtrain,ytrain,epochs=20, batch_size=5, verbose=True,validation_data=(xte
Epoch 1/20
1054/1054 [=====] - 399s 378ms/step - loss: 0.2523 - accuracy:
Epoch 2/20
1054/1054 [=====] - 398s 377ms/step - loss: 0.1383 - accuracy:
Epoch 3/20
1054/1054 [=====] - 398s 378ms/step - loss: 0.1179 - accuracy:
Epoch 4/20
1054/1054 [=====] - 398s 377ms/step - loss: 0.1000 - accuracy:
```

```
Epoch 5/20
1054/1054 [=====] - 398s 377ms/step - loss: 0.0873 - accuracy: 0.9129
Epoch 6/20
1054/1054 [=====] - 398s 377ms/step - loss: 0.0736 - accuracy: 0.9171
Epoch 7/20
1054/1054 [=====] - 398s 378ms/step - loss: 0.0653 - accuracy: 0.9213
Epoch 8/20
1054/1054 [=====] - 397s 377ms/step - loss: 0.0587 - accuracy: 0.9245
Epoch 9/20
1054/1054 [=====] - 397s 377ms/step - loss: 0.0459 - accuracy: 0.9307
Epoch 10/20
1054/1054 [=====] - 397s 377ms/step - loss: 0.0400 - accuracy: 0.9348
Epoch 11/20
1054/1054 [=====] - 398s 377ms/step - loss: 0.0346 - accuracy: 0.9389
Epoch 12/20
1054/1054 [=====] - 402s 382ms/step - loss: 0.0251 - accuracy: 0.9429
Epoch 13/20
1054/1054 [=====] - 398s 377ms/step - loss: 0.0216 - accuracy: 0.9469
Epoch 14/20
1054/1054 [=====] - 398s 378ms/step - loss: 0.0261 - accuracy: 0.9499
Epoch 15/20
1054/1054 [=====] - 398s 378ms/step - loss: 0.0193 - accuracy: 0.9539
Epoch 16/20
1054/1054 [=====] - 398s 377ms/step - loss: 0.0220 - accuracy: 0.9569
Epoch 17/20
1054/1054 [=====] - 397s 377ms/step - loss: 0.0174 - accuracy: 0.9609
Epoch 18/20
1054/1054 [=====] - 398s 377ms/step - loss: 0.0128 - accuracy: 0.9649
Epoch 19/20
1054/1054 [=====] - 398s 378ms/step - loss: 0.0132 - accuracy: 0.9679
Epoch 20/20
1054/1054 [=====] - 397s 377ms/step - loss: 0.0165 - accuracy: 0.9709
```



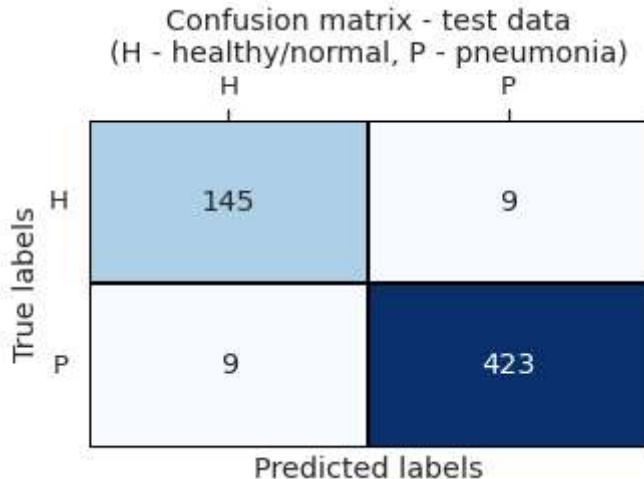
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)

report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyplot.py'>
precision    recall    f1-score   support
          0       0.94      0.94      0.94      154
          1       0.98      0.98      0.98      432
   accuracy                           0.97      586
  macro avg       0.96      0.96      0.96      586
weighted avg       0.97      0.97      0.97      586
```

## Batch size 100 + Lr 0.0001 + Epoch 10

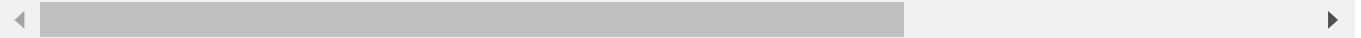
```
model = Vgg16.build()
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.0001),metrics
history = model.fit(xtrain,ytrain,epochs=10, batch_size=100, verbose=True,validation_data=(xt

Epoch 1/10
53/53 [=====] - 156s 2s/step - loss: 0.5896 - accuracy: 0.7767
Epoch 2/10
53/53 [=====] - 103s 2s/step - loss: 0.1692 - accuracy: 0.9343
Epoch 3/10
53/53 [=====] - 104s 2s/step - loss: 0.1315 - accuracy: 0.9514
Epoch 4/10
53/53 [=====] - 104s 2s/step - loss: 0.1167 - accuracy: 0.9560
```

```

Epoch 5/10
53/53 [=====] - 104s 2s/step - loss: 0.1069 - accuracy: 0.9602
Epoch 6/10
53/53 [=====] - 104s 2s/step - loss: 0.1079 - accuracy: 0.9602
Epoch 7/10
53/53 [=====] - 104s 2s/step - loss: 0.0894 - accuracy: 0.9670
Epoch 8/10
53/53 [=====] - 104s 2s/step - loss: 0.0874 - accuracy: 0.9670
Epoch 9/10
53/53 [=====] - 104s 2s/step - loss: 0.0718 - accuracy: 0.9732
Epoch 10/10
53/53 [=====] - 104s 2s/step - loss: 0.0585 - accuracy: 0.9761

```

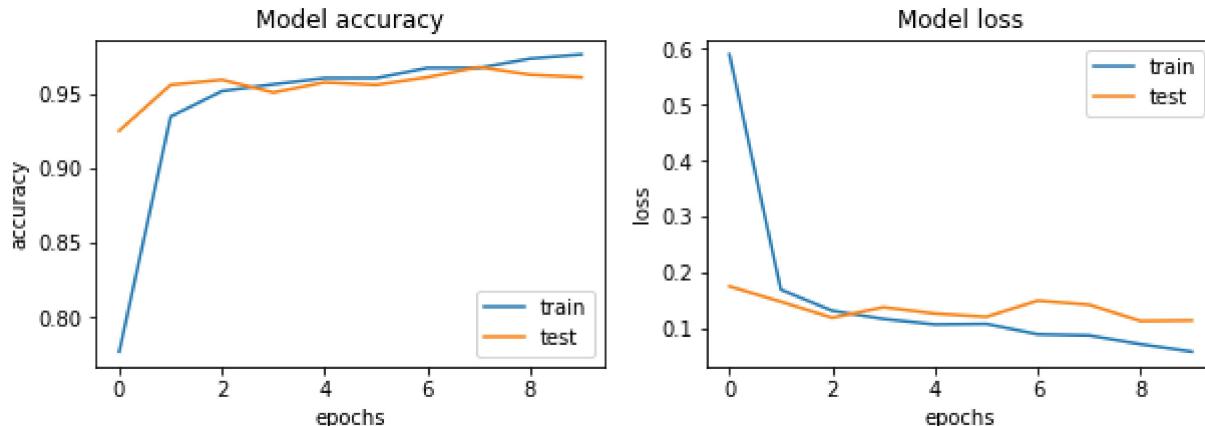


```

# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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

```



```

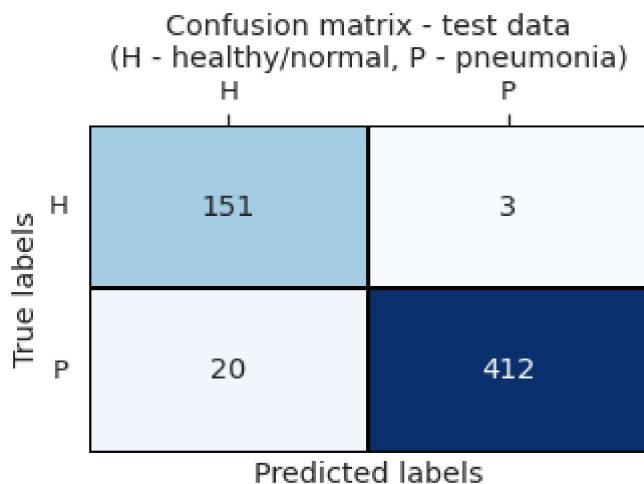
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)

```

```

report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)

```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision    recall    f1-score    support
0            0.88      0.98      0.93      154
1            0.99      0.95      0.97      432
```

## Batch size 100 + Lr 0.0001 + Epoch 10 + Probability 0.2

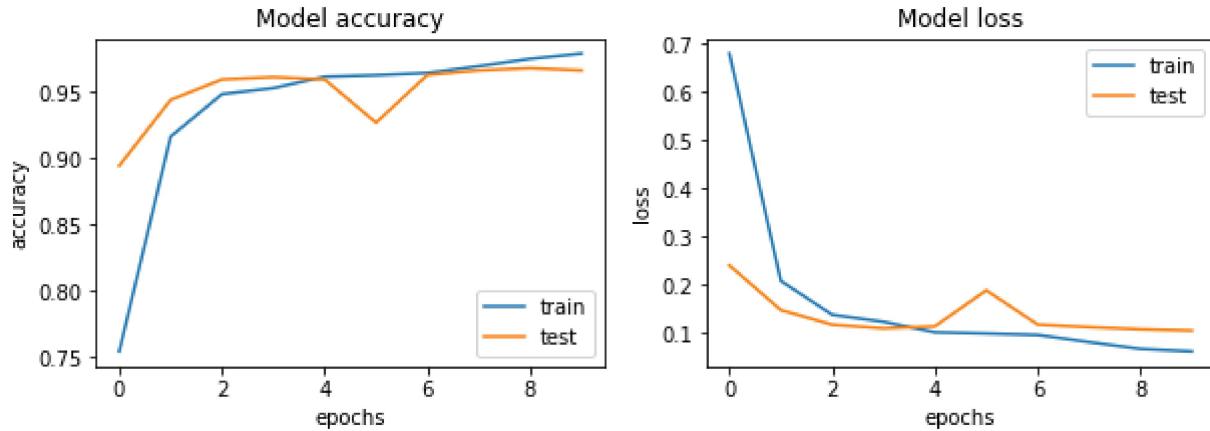
```
weighted avg      0.96      0.96      0.96      586
```

```
model = Vgg16.build_dropout(0.2)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.0001),metrics
history = model.fit(xtrain,ytrain,epochs=10, batch_size=100, verbose=True,validation_data=(xt

Epoch 1/10
53/53 [=====] - 165s 3s/step - loss: 0.6798 - accuracy: 0.7546
Epoch 2/10
53/53 [=====] - 110s 2s/step - loss: 0.2074 - accuracy: 0.9161
Epoch 3/10
53/53 [=====] - 111s 2s/step - loss: 0.1371 - accuracy: 0.9480
Epoch 4/10
53/53 [=====] - 111s 2s/step - loss: 0.1224 - accuracy: 0.9526
Epoch 5/10
53/53 [=====] - 111s 2s/step - loss: 0.1006 - accuracy: 0.9611
Epoch 6/10
53/53 [=====] - 111s 2s/step - loss: 0.0983 - accuracy: 0.9622
Epoch 7/10
53/53 [=====] - 111s 2s/step - loss: 0.0953 - accuracy: 0.9639
Epoch 8/10
53/53 [=====] - 111s 2s/step - loss: 0.0804 - accuracy: 0.9691
Epoch 9/10
53/53 [=====] - 111s 2s/step - loss: 0.0663 - accuracy: 0.9746
Epoch 10/10
53/53 [=====] - 111s 2s/step - loss: 0.0611 - accuracy: 0.9786
```

```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()
```

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)  
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)  
print(report_normal)
```

## Batch size 100 + Lr 0.0001 + Epoch 10 + Probability 0.5

```
model = Vgg16.build_dropout(0.5)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.0001),metrics
history = model.fit(xtrain,ytrain,epochs=10, batch_size=100, verbose=True,validation_data=(xt

Epoch 1/10
53/53 [=====] - 152s 3s/step - loss: 0.4906 - accuracy: 0.8091
Epoch 2/10
53/53 [=====] - 150s 3s/step - loss: 0.1779 - accuracy: 0.9324
Epoch 3/10
53/53 [=====] - 150s 3s/step - loss: 0.1348 - accuracy: 0.9471
Epoch 4/10
53/53 [=====] - 150s 3s/step - loss: 0.1176 - accuracy: 0.9567
Epoch 5/10
53/53 [=====] - 150s 3s/step - loss: 0.1034 - accuracy: 0.9598
Epoch 6/10
53/53 [=====] - 150s 3s/step - loss: 0.0874 - accuracy: 0.9679
Epoch 7/10
53/53 [=====] - 150s 3s/step - loss: 0.0842 - accuracy: 0.9683
Epoch 8/10
53/53 [=====] - 150s 3s/step - loss: 0.0737 - accuracy: 0.9725
Epoch 9/10
53/53 [=====] - 150s 3s/step - loss: 0.0764 - accuracy: 0.9702
Epoch 10/10
53/53 [=====] - 150s 3s/step - loss: 0.0625 - accuracy: 0.9769
```

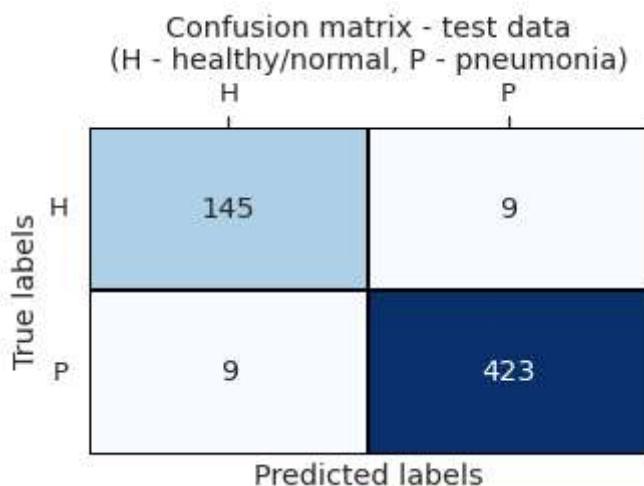
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision    recall    f1-score    support
0            0.94     0.94     0.94      154
1            0.98     0.98     0.98      432
accuracy                           0.97      586
macro avg       0.96     0.96     0.96      586
weighted avg    0.97     0.97     0.97      586
```



## Batch size 100 + Lr 0.0001 + Epoch 20

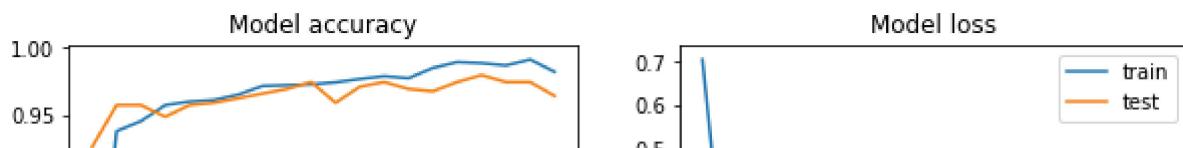
```
model = Vgg16.build()
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.0001),metrics
history = model.fit(xtrain,ytrain,epochs=20, batch_size=100, verbose=True,validation_data=(xt
Epoch 1/20
53/53 [=====] - 156s 2s/step - loss: 0.7055 - accuracy: 0.7708
Epoch 2/20
53/53 [=====] - 106s 2s/step - loss: 0.1617 - accuracy: 0.9381
Epoch 3/20
53/53 [=====] - 109s 2s/step - loss: 0.1384 - accuracy: 0.9455
Epoch 4/20
53/53 [=====] - 111s 2s/step - loss: 0.1164 - accuracy: 0.9573
Epoch 5/20
```

```
53/53 [=====] - 111s 2s/step - loss: 0.1046 - accuracy: 0.9600
Epoch 6/20
53/53 [=====] - 111s 2s/step - loss: 0.1016 - accuracy: 0.9611
Epoch 7/20
53/53 [=====] - 111s 2s/step - loss: 0.0951 - accuracy: 0.9651
Epoch 8/20
53/53 [=====] - 111s 2s/step - loss: 0.0774 - accuracy: 0.9715
Epoch 9/20
53/53 [=====] - 111s 2s/step - loss: 0.0805 - accuracy: 0.9721
Epoch 10/20
53/53 [=====] - 111s 2s/step - loss: 0.0695 - accuracy: 0.9727
Epoch 11/20
53/53 [=====] - 111s 2s/step - loss: 0.0664 - accuracy: 0.9742
Epoch 12/20
53/53 [=====] - 111s 2s/step - loss: 0.0569 - accuracy: 0.9767
Epoch 13/20
53/53 [=====] - 111s 2s/step - loss: 0.0561 - accuracy: 0.9787
Epoch 14/20
53/53 [=====] - 111s 2s/step - loss: 0.0552 - accuracy: 0.9772
Epoch 15/20
53/53 [=====] - 111s 2s/step - loss: 0.0368 - accuracy: 0.9848
Epoch 16/20
53/53 [=====] - 111s 2s/step - loss: 0.0297 - accuracy: 0.9892
Epoch 17/20
53/53 [=====] - 111s 2s/step - loss: 0.0286 - accuracy: 0.9884
Epoch 18/20
53/53 [=====] - 111s 2s/step - loss: 0.0333 - accuracy: 0.9867
Epoch 19/20
53/53 [=====] - 111s 2s/step - loss: 0.0214 - accuracy: 0.9911
Epoch 20/20
53/53 [=====] - 111s 2s/step - loss: 0.0504 - accuracy: 0.9820
```



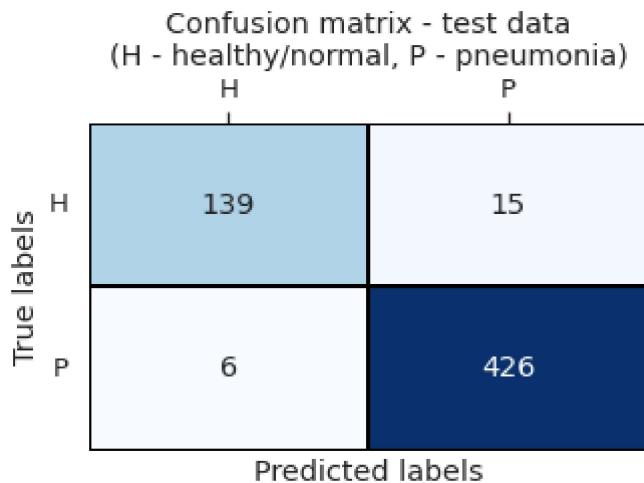
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)

report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision      recall    f1-score   support
0            0.96     0.90      0.93     154
1            0.97     0.99      0.98     432
accuracy                           0.96     586
macro avg       0.96     0.94      0.95     586
weighted avg    0.96     0.96      0.96     586
```



Batch size 100 + Lr 0.0001 + Epoch 20 + Probability 0.2

```
model = Vgg16.build_dropout(0.2)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.0001),metrics
history = model.fit(xtrain,ytrain,epochs=20, batch_size=100, verbose=True,validation_data=(xt
Epoch 1/20
53/53 [=====] - 152s 3s/step - loss: 0.6204 - accuracy: 0.7789
Epoch 2/20
53/53 [=====] - 150s 3s/step - loss: 0.1863 - accuracy: 0.9275
Epoch 3/20
53/53 [=====] - 150s 3s/step - loss: 0.1361 - accuracy: 0.9457
Epoch 4/20
53/53 [=====] - 150s 3s/step - loss: 0.1256 - accuracy: 0.9545
```

```
Epoch 5/20
53/53 [=====] - 150s 3s/step - loss: 0.1070 - accuracy: 0.9605
Epoch 6/20
53/53 [=====] - 150s 3s/step - loss: 0.0923 - accuracy: 0.9653
Epoch 7/20
53/53 [=====] - 151s 3s/step - loss: 0.0787 - accuracy: 0.9689
Epoch 8/20
53/53 [=====] - 151s 3s/step - loss: 0.0808 - accuracy: 0.9691
Epoch 9/20
53/53 [=====] - 150s 3s/step - loss: 0.0726 - accuracy: 0.9744
Epoch 10/20
53/53 [=====] - 150s 3s/step - loss: 0.0673 - accuracy: 0.9744
Epoch 11/20
53/53 [=====] - 150s 3s/step - loss: 0.0544 - accuracy: 0.9793
Epoch 12/20
53/53 [=====] - 151s 3s/step - loss: 0.0447 - accuracy: 0.9827
Epoch 13/20
53/53 [=====] - 151s 3s/step - loss: 0.0450 - accuracy: 0.9835
Epoch 14/20
53/53 [=====] - 150s 3s/step - loss: 0.0560 - accuracy: 0.9805
Epoch 15/20
53/53 [=====] - 150s 3s/step - loss: 0.0452 - accuracy: 0.9814
Epoch 16/20
53/53 [=====] - 150s 3s/step - loss: 0.0387 - accuracy: 0.9844
Epoch 17/20
53/53 [=====] - 150s 3s/step - loss: 0.0323 - accuracy: 0.9877
Epoch 18/20
53/53 [=====] - 150s 3s/step - loss: 0.0245 - accuracy: 0.9903
Epoch 19/20
53/53 [=====] - 150s 3s/step - loss: 0.0470 - accuracy: 0.9846
Epoch 20/20
53/53 [=====] - 150s 3s/step - loss: 0.0341 - accuracy: 0.9869
```



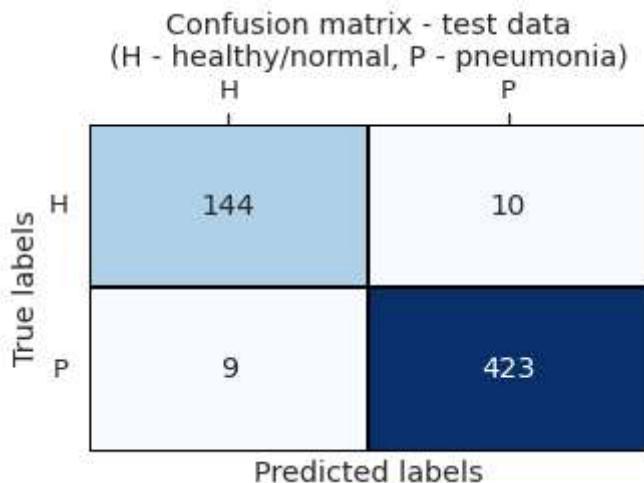
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyplot.py'>
precision    recall    f1-score   support
          0       0.94      0.94      0.94      154
          1       0.98      0.98      0.98      432
   accuracy                           0.97      586
  macro avg       0.96      0.96      0.96      586
weighted avg       0.97      0.97      0.97      586
```

Batch size 100 + Lr 0.0001 + Epoch 20 + Probability 0.5

```
model = Vgg16.build_dropout(0.5)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.0001),metrics
history = model.fit(xtrain,ytrain,epochs=20, batch_size=100, verbose=True,validation_data=(xt
```

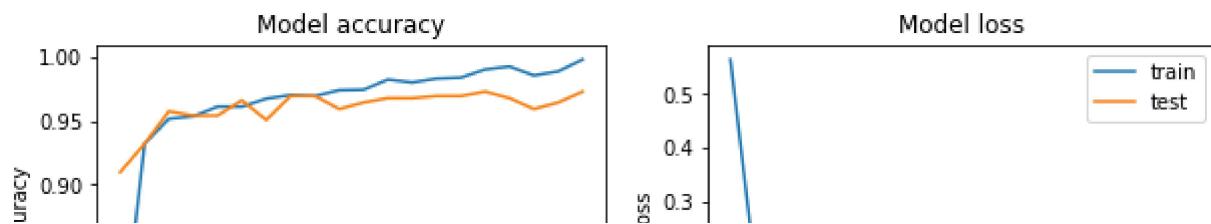
```
Epoch 1/20
53/53 [=====] - 163s 2s/step - loss: 0.5641 - accuracy: 0.7641
Epoch 2/20
53/53 [=====] - 111s 2s/step - loss: 0.1795 - accuracy: 0.9319
Epoch 3/20
53/53 [=====] - 114s 2s/step - loss: 0.1376 - accuracy: 0.9514
Epoch 4/20
```

```
53/53 [=====] - 115s 2s/step - loss: 0.1268 - accuracy: 0.9533
Epoch 5/20
53/53 [=====] - 116s 2s/step - loss: 0.1150 - accuracy: 0.9609
Epoch 6/20
53/53 [=====] - 116s 2s/step - loss: 0.1032 - accuracy: 0.9609
Epoch 7/20
53/53 [=====] - 116s 2s/step - loss: 0.0872 - accuracy: 0.9672
Epoch 8/20
53/53 [=====] - 116s 2s/step - loss: 0.0748 - accuracy: 0.9700
Epoch 9/20
53/53 [=====] - 116s 2s/step - loss: 0.0721 - accuracy: 0.9693
Epoch 10/20
53/53 [=====] - 116s 2s/step - loss: 0.0669 - accuracy: 0.9738
Epoch 11/20
53/53 [=====] - 116s 2s/step - loss: 0.0600 - accuracy: 0.9742
Epoch 12/20
53/53 [=====] - 116s 2s/step - loss: 0.0483 - accuracy: 0.9822
Epoch 13/20
53/53 [=====] - 115s 2s/step - loss: 0.0525 - accuracy: 0.9799
Epoch 14/20
53/53 [=====] - 115s 2s/step - loss: 0.0471 - accuracy: 0.9829
Epoch 15/20
53/53 [=====] - 115s 2s/step - loss: 0.0388 - accuracy: 0.9837
Epoch 16/20
53/53 [=====] - 115s 2s/step - loss: 0.0290 - accuracy: 0.9901
Epoch 17/20
53/53 [=====] - 115s 2s/step - loss: 0.0216 - accuracy: 0.9924
Epoch 18/20
53/53 [=====] - 115s 2s/step - loss: 0.0397 - accuracy: 0.9854
Epoch 19/20
53/53 [=====] - 115s 2s/step - loss: 0.0298 - accuracy: 0.9886
Epoch 20/20
53/53 [=====] - 115s 2s/step - loss: 0.0081 - accuracy: 0.9979
```



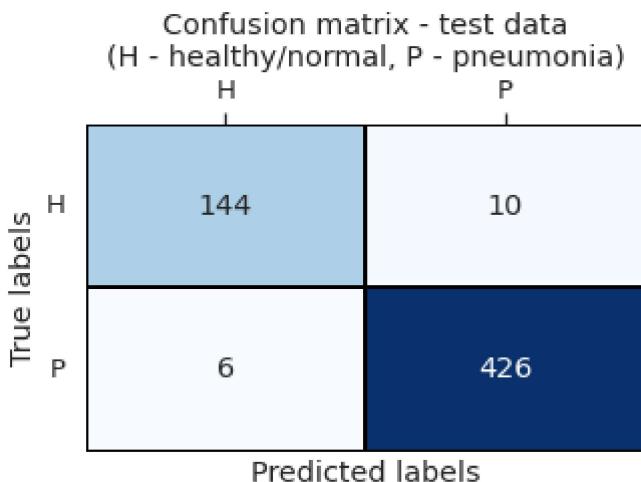
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)

report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyplot.py'>
precision    recall    f1-score    support
          0       0.96      0.94      0.95      154
          1       0.98      0.99      0.98      432
   accuracy                           0.97      586
  macro avg       0.97      0.96      0.96      586
weighted avg       0.97      0.97      0.97      586
```



## Batch size 100 + Lr 0.00001 + Epoch 10

```
model = Vgg16.build()
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.00001),metrics=['accuracy'])
history = model.fit(xtrain,ytrain,epochs=10, batch_size=100, verbose=True,validation_data=(xt,yt))

Epoch 1/10
53/53 [=====] - 157s 3s/step - loss: 0.3858 - accuracy: 0.8288
Epoch 2/10
53/53 [=====] - 157s 3s/step - loss: 0.1509 - accuracy: 0.9419
Epoch 3/10
53/53 [=====] - 158s 3s/step - loss: 0.1272 - accuracy: 0.9528
```

```

Epoch 4/10
53/53 [=====] - 158s 3s/step - loss: 0.1121 - accuracy: 0.9588
Epoch 5/10
53/53 [=====] - 158s 3s/step - loss: 0.1116 - accuracy: 0.9590
Epoch 6/10
53/53 [=====] - 158s 3s/step - loss: 0.0923 - accuracy: 0.9651
Epoch 7/10
53/53 [=====] - 158s 3s/step - loss: 0.0782 - accuracy: 0.9706
Epoch 8/10
53/53 [=====] - 158s 3s/step - loss: 0.0851 - accuracy: 0.9694
Epoch 9/10
53/53 [=====] - 158s 3s/step - loss: 0.0706 - accuracy: 0.9774
Epoch 10/10
53/53 [=====] - 158s 3s/step - loss: 0.0623 - accuracy: 0.9774

```

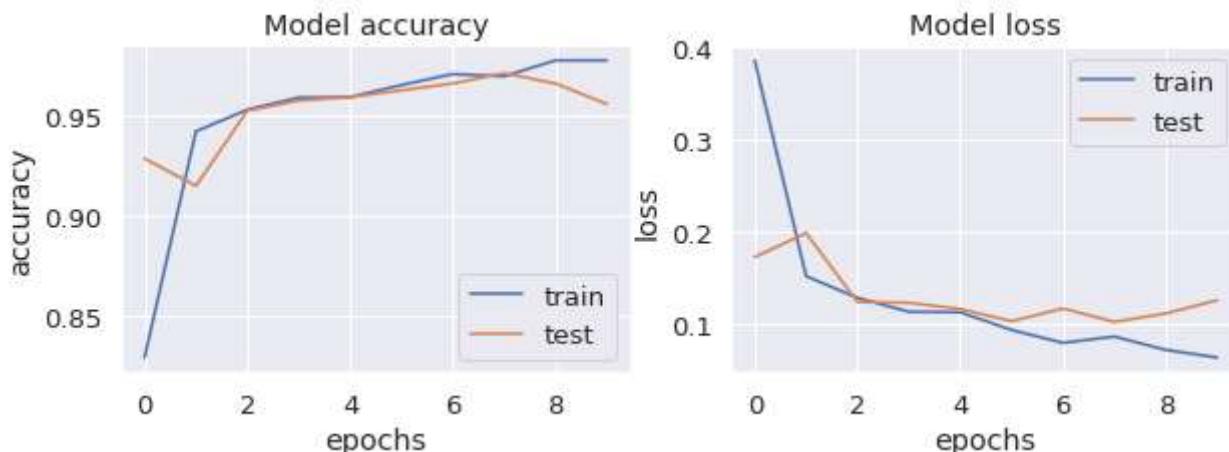


```

# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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

```



```

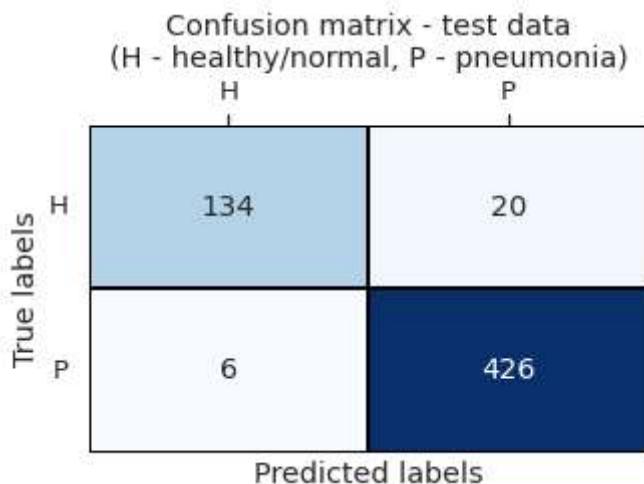
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)

```

```

report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)

```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision      recall   f1-score   support
          0       0.96      0.87      0.91      154
          1       0.96      0.99      0.97      432
  accuracy                           0.96      586
  macro avg       0.96      0.92      0.91      596
```

Batch size 100 + Lr 0.00001 + Epoch 10 + Probability 0.2

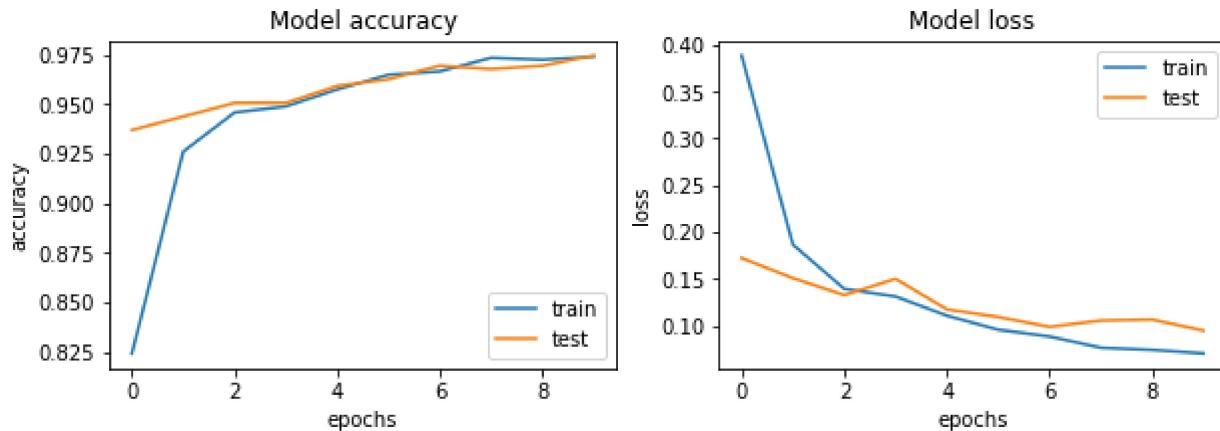
```
model = Vgg16.build_dropout(0.2)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.00001),metric
history = model.fit(xtrain,ytrain,epochs=10, batch_size=100, verbose=True,validation_data=(xt

Epoch 1/10
53/53 [=====] - 162s 2s/step - loss: 0.3887 - accuracy: 0.8243
Epoch 2/10
53/53 [=====] - 107s 2s/step - loss: 0.1864 - accuracy: 0.9260
Epoch 3/10
53/53 [=====] - 107s 2s/step - loss: 0.1391 - accuracy: 0.9457
Epoch 4/10
53/53 [=====] - 107s 2s/step - loss: 0.1310 - accuracy: 0.9488
Epoch 5/10
53/53 [=====] - 107s 2s/step - loss: 0.1105 - accuracy: 0.9573
Epoch 6/10
53/53 [=====] - 107s 2s/step - loss: 0.0954 - accuracy: 0.9647
Epoch 7/10
53/53 [=====] - 107s 2s/step - loss: 0.0882 - accuracy: 0.9664
Epoch 8/10
53/53 [=====] - 107s 2s/step - loss: 0.0763 - accuracy: 0.9732
Epoch 9/10
53/53 [=====] - 107s 2s/step - loss: 0.0739 - accuracy: 0.9723
Epoch 10/10
53/53 [=====] - 107s 2s/step - loss: 0.0702 - accuracy: 0.9738
```

# Visualize Loss and Accuracy Rates

```
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```

Confusion matrix - test data  
(H - healthy/normal, P - pneumonia)

	H	P
H	True Negatives	False Positives
P	False Negatives	True Positives

Batch size 100 + Lr 0.00001 + Epoch 10 + Probability 0.5

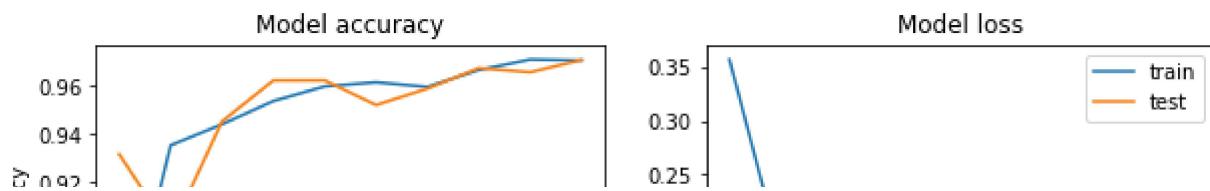
```
model = Vgg16.build_dropout(0.5)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.00001),metric
history = model.fit(xtrain,ytrain,epochs=10, batch_size=100, verbose=True,validation_data=(xt

Epoch 1/10
53/53 [=====] - 162s 2s/step - loss: 0.3567 - accuracy: 0.8472
Epoch 2/10
53/53 [=====] - 113s 2s/step - loss: 0.1753 - accuracy: 0.9355
Epoch 3/10
53/53 [=====] - 115s 2s/step - loss: 0.1460 - accuracy: 0.9442
Epoch 4/10
53/53 [=====] - 114s 2s/step - loss: 0.1215 - accuracy: 0.9539
Epoch 5/10
53/53 [=====] - 115s 2s/step - loss: 0.1086 - accuracy: 0.9600
Epoch 6/10
53/53 [=====] - 115s 2s/step - loss: 0.1006 - accuracy: 0.9617
Epoch 7/10
53/53 [=====] - 114s 2s/step - loss: 0.1020 - accuracy: 0.9598
Epoch 8/10
53/53 [=====] - 115s 2s/step - loss: 0.0908 - accuracy: 0.9668
Epoch 9/10
53/53 [=====] - 115s 2s/step - loss: 0.0802 - accuracy: 0.9712
Epoch 10/10
53/53 [=====] - 115s 2s/step - loss: 0.0825 - accuracy: 0.9708
```



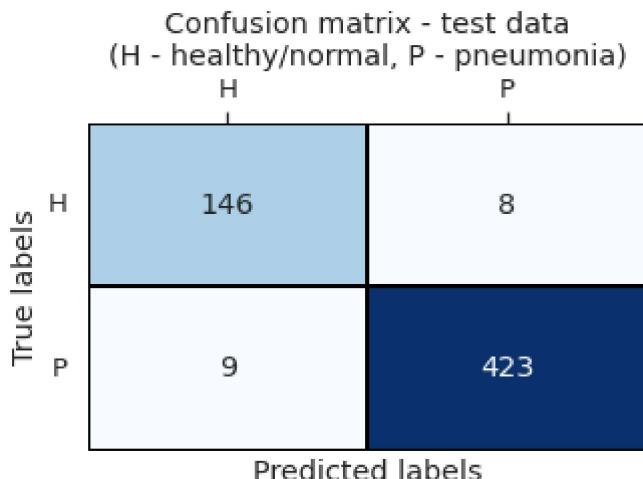
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision      recall    f1-score    support
          0       0.94      0.95      0.94      154
          1       0.98      0.98      0.98      432
   accuracy                           0.97      586
  macro avg       0.96      0.96      0.96      586
weighted avg       0.97      0.97      0.97      586
```

Batch size 100 + Lr 0.00001 + Epoch 20

```
model = Vgg16.build()
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.00001))
history = model.fit(xtrain,ytrain,epochs=20, batch_size=100, verbose=True, validation_d
```

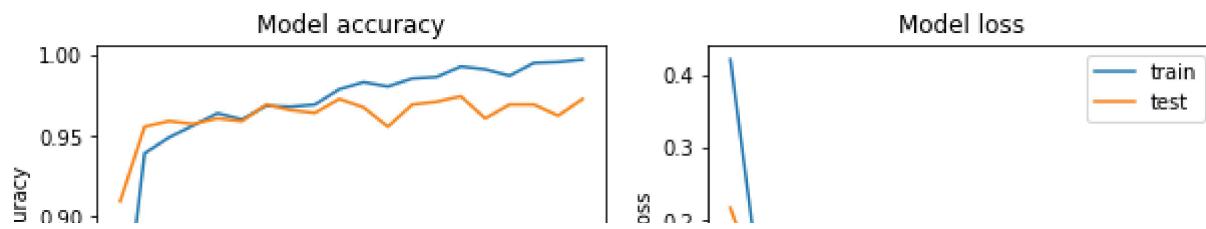
```
Epoch 1/20
53/53 [=====] - 165s 3s/step - loss: 0.4205 - accuracy: 0.8129
Epoch 2/20
53/53 [=====] - 111s 2s/step - loss: 0.1656 - accuracy: 0.9391
Epoch 3/20
53/53 [=====] - 112s 2s/step - loss: 0.1330 - accuracy: 0.9490
Epoch 4/20
```

```
53/53 [=====] - 112s 2s/step - loss: 0.1165 - accuracy: 0.9564
Epoch 5/20
53/53 [=====] - 112s 2s/step - loss: 0.1026 - accuracy: 0.9639
Epoch 6/20
53/53 [=====] - 112s 2s/step - loss: 0.1051 - accuracy: 0.9602
Epoch 7/20
53/53 [=====] - 113s 2s/step - loss: 0.0851 - accuracy: 0.9685
Epoch 8/20
53/53 [=====] - 112s 2s/step - loss: 0.0815 - accuracy: 0.9679
Epoch 9/20
53/53 [=====] - 112s 2s/step - loss: 0.0792 - accuracy: 0.9693
Epoch 10/20
53/53 [=====] - 112s 2s/step - loss: 0.0637 - accuracy: 0.9787
Epoch 11/20
53/53 [=====] - 112s 2s/step - loss: 0.0507 - accuracy: 0.9831
Epoch 12/20
53/53 [=====] - 113s 2s/step - loss: 0.0545 - accuracy: 0.9805
Epoch 13/20
53/53 [=====] - 113s 2s/step - loss: 0.0407 - accuracy: 0.9854
Epoch 14/20
53/53 [=====] - 112s 2s/step - loss: 0.0366 - accuracy: 0.9863
Epoch 15/20
53/53 [=====] - 113s 2s/step - loss: 0.0244 - accuracy: 0.9928
Epoch 16/20
53/53 [=====] - 113s 2s/step - loss: 0.0255 - accuracy: 0.9911
Epoch 17/20
53/53 [=====] - 113s 2s/step - loss: 0.0341 - accuracy: 0.9871
Epoch 18/20
53/53 [=====] - 113s 2s/step - loss: 0.0164 - accuracy: 0.9951
Epoch 19/20
53/53 [=====] - 113s 2s/step - loss: 0.0140 - accuracy: 0.9956
Epoch 20/20
53/53 [=====] - 113s 2s/step - loss: 0.0115 - accuracy: 0.9972
```



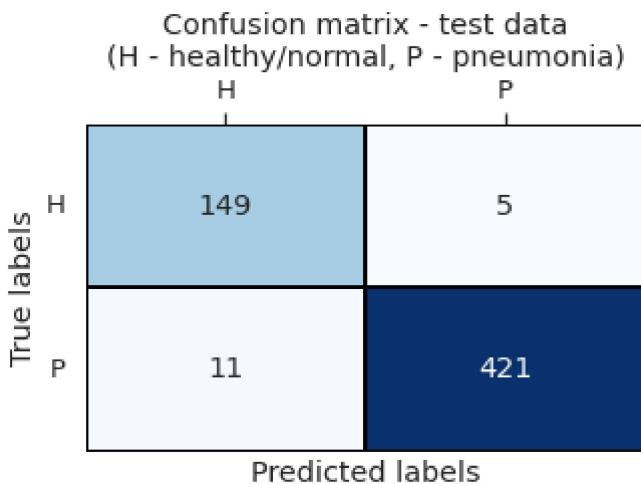
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)

report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision    recall    f1-score    support
          0       0.93      0.97      0.95      154
          1       0.99      0.97      0.98      432
   accuracy                           0.97      586
  macro avg       0.96      0.97      0.97      586
weighted avg       0.97      0.97      0.97      586
```



## Batch size 100 + Lr 0.00001 + Epoch 20 + Probability 0.2

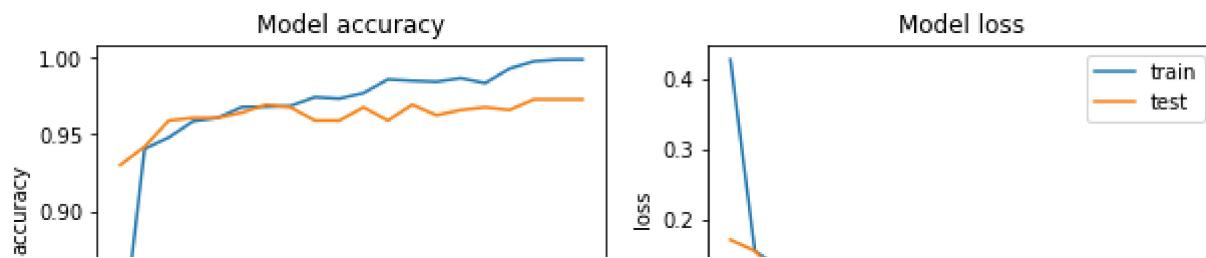
```
model = Vgg16.build_dropout(0.2)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.00001),metric
history = model.fit(xtrain,ytrain,epochs=20, batch_size=100, verbose=True,validation_data=(xt
Epoch 1/20
53/53 [=====] - 164s 3s/step - loss: 0.4272 - accuracy: 0.8049
Epoch 2/20
53/53 [=====] - 111s 2s/step - loss: 0.1564 - accuracy: 0.9406
Epoch 3/20
53/53 [=====] - 112s 2s/step - loss: 0.1315 - accuracy: 0.9480
```

```
Epoch 4/20
53/53 [=====] - 112s 2s/step - loss: 0.1126 - accuracy: 0.9586
Epoch 5/20
53/53 [=====] - 112s 2s/step - loss: 0.1029 - accuracy: 0.9607
Epoch 6/20
53/53 [=====] - 112s 2s/step - loss: 0.0894 - accuracy: 0.9677
Epoch 7/20
53/53 [=====] - 112s 2s/step - loss: 0.0882 - accuracy: 0.9679
Epoch 8/20
53/53 [=====] - 112s 2s/step - loss: 0.0844 - accuracy: 0.9685
Epoch 9/20
53/53 [=====] - 113s 2s/step - loss: 0.0707 - accuracy: 0.9742
Epoch 10/20
53/53 [=====] - 113s 2s/step - loss: 0.0689 - accuracy: 0.9732
Epoch 11/20
53/53 [=====] - 113s 2s/step - loss: 0.0637 - accuracy: 0.9769
Epoch 12/20
53/53 [=====] - 113s 2s/step - loss: 0.0438 - accuracy: 0.9858
Epoch 13/20
53/53 [=====] - 113s 2s/step - loss: 0.0449 - accuracy: 0.9848
Epoch 14/20
53/53 [=====] - 112s 2s/step - loss: 0.0435 - accuracy: 0.9843
Epoch 15/20
53/53 [=====] - 113s 2s/step - loss: 0.0353 - accuracy: 0.9865
Epoch 16/20
53/53 [=====] - 113s 2s/step - loss: 0.0454 - accuracy: 0.9833
Epoch 17/20
53/53 [=====] - 113s 2s/step - loss: 0.0219 - accuracy: 0.9928
Epoch 18/20
53/53 [=====] - 113s 2s/step - loss: 0.0116 - accuracy: 0.9975
Epoch 19/20
53/53 [=====] - 113s 2s/step - loss: 0.0072 - accuracy: 0.9987
Epoch 20/20
53/53 [=====] - 113s 2s/step - loss: 0.0054 - accuracy: 0.9987
```



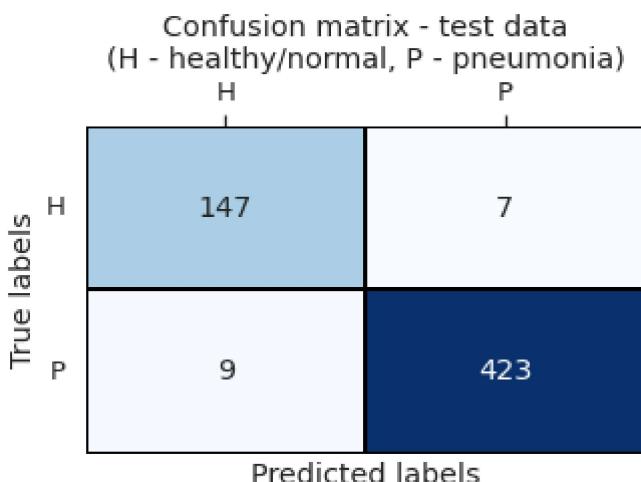
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```

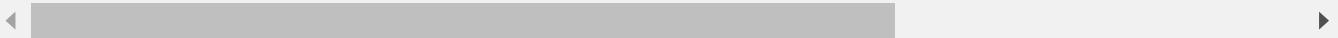


```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
precision    recall    f1-score    support
      0          0.94      0.95      0.95      154
      1          0.98      0.98      0.98      432
   accuracy                           0.97      586
  macro avg       0.96      0.97      0.96      586
weighted avg       0.97      0.97      0.97      586
```

Batch size 100 + Lr 0.00001 + Epoch 20 + Probability 0.5

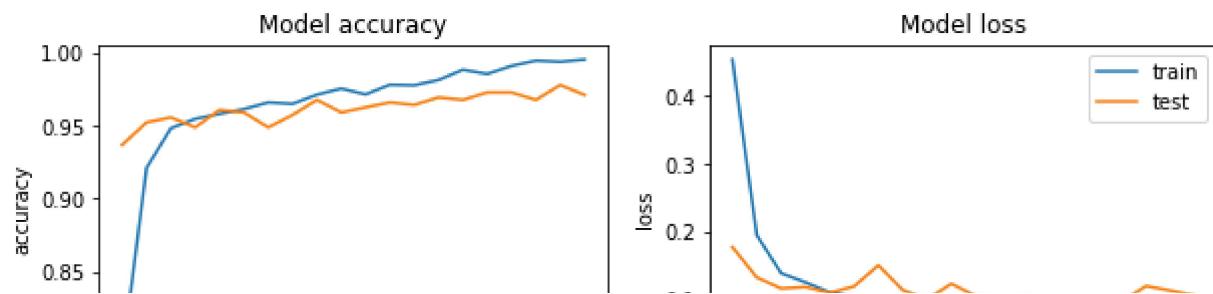
```
model = Vgg16.build_dropout(0.5)
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate = 0.00001),metric
history = model.fit(xtrain,ytrain,epochs=20, batch_size=100, verbose=True,validation_data=(xt
Epoch 1/20
53/53 [=====] - 170s 3s/step - loss: 0.4534 - accuracy: 0.7913
Epoch 2/20
53/53 [=====] - 119s 2s/step - loss: 0.1949 - accuracy: 0.9211
Epoch 3/20
```

```
53/53 [=====] - 119s 2s/step - loss: 0.1391 - accuracy: 0.9482
Epoch 4/20
53/53 [=====] - 120s 2s/step - loss: 0.1256 - accuracy: 0.9546
Epoch 5/20
53/53 [=====] - 119s 2s/step - loss: 0.1107 - accuracy: 0.9581
Epoch 6/20
53/53 [=====] - 119s 2s/step - loss: 0.1039 - accuracy: 0.9613
Epoch 7/20
53/53 [=====] - 120s 2s/step - loss: 0.0957 - accuracy: 0.9658
Epoch 8/20
53/53 [=====] - 119s 2s/step - loss: 0.0908 - accuracy: 0.9649
Epoch 9/20
53/53 [=====] - 120s 2s/step - loss: 0.0793 - accuracy: 0.9712
Epoch 10/20
53/53 [=====] - 120s 2s/step - loss: 0.0711 - accuracy: 0.9753
Epoch 11/20
53/53 [=====] - 120s 2s/step - loss: 0.0736 - accuracy: 0.9713
Epoch 12/20
53/53 [=====] - 120s 2s/step - loss: 0.0604 - accuracy: 0.9780
Epoch 13/20
53/53 [=====] - 120s 2s/step - loss: 0.0615 - accuracy: 0.9776
Epoch 14/20
53/53 [=====] - 120s 2s/step - loss: 0.0514 - accuracy: 0.9812
Epoch 15/20
53/53 [=====] - 120s 2s/step - loss: 0.0376 - accuracy: 0.9882
Epoch 16/20
53/53 [=====] - 120s 2s/step - loss: 0.0376 - accuracy: 0.9854
Epoch 17/20
53/53 [=====] - 120s 2s/step - loss: 0.0295 - accuracy: 0.9909
Epoch 18/20
53/53 [=====] - 120s 2s/step - loss: 0.0199 - accuracy: 0.9945
Epoch 19/20
53/53 [=====] - 120s 2s/step - loss: 0.0209 - accuracy: 0.9937
Epoch 20/20
53/53 [=====] - 120s 2s/step - loss: 0.0153 - accuracy: 0.9953
```



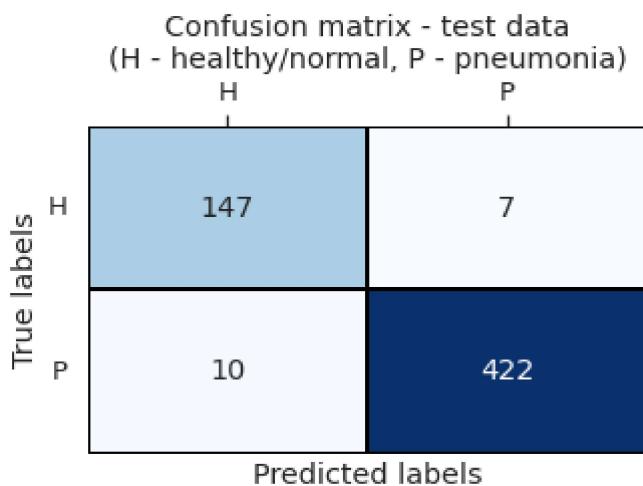
```
# Visualize Loss and Accuracy Rates
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
ax = ax.ravel()

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



```
heatmap_normal = ConfMatrix.heatmap(xtest, ytest)
print(heatmap_normal)
```

```
report_normal = ConfMatrix.classification_report(ytest)
print(report_normal)
```



```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.7/dist-packages/matplotlib/pyp]
```

	precision	recall	f1-score	support
0	0.94	0.95	0.95	154
1	0.98	0.98	0.98	432
accuracy			0.97	586
macro avg	0.96	0.97	0.96	586
weighted avg	0.97	0.97	0.97	586

	precision	recall	f1-score	support
0	0.94	0.95	0.95	154
1	0.98	0.98	0.98	432
accuracy			0.97	586
macro avg	0.96	0.97	0.96	586
weighted avg	0.97	0.97	0.97	586

---

✓ 0s completed at 1:12 PM

