johnlocke333 /
xray_image_classification

Code Issues Pull requests Actions Projects Wiki Security

xray_image_classification / notebook.ipynb

johnlocke333 Evaluation, added more information about the models and gridsearch re...

0d62c94 · 5 minutes ago

3017 lines (3017 loc) · 507 KB

Preview Code Blame Raw

Open in Colab

Final Project Submission

- Student name: Jack Locke
- Student pace: Self-paced
- Instructor name: Mark Barbour

Pneumonia Model Analysis

Overview

1. Business Understanding
2. Data Understanding
3. Data Preparation

- Directory path for train/test/valid images
- Create image/label datasets and reshape
- Change dimensions of images and labels
- Standardize images
- Class Imbalance - class weights

4. Modeling

- Baseline
- Gridsearch/Tuned Model

Business Understanding

The business stakeholder is a diagnostic imaging center exploring the use of neural networks to help predict when patients have pneumonia. My project uses X-ray images of patients with and without pneumonia. The model aims to predict whether someone has pneumonia or not. The importance of the model revolves around the costs associated with false positives and false negatives. The false positive would be predicting someone does have pneumonia when they don’t (wasted time/resources/money). The false negative would be predicting someone didn't have pneumonia when they did (future health issues). I will find an appropriate trade-off between our two costs, focusing on minimizing future health issues. The imaging center can use the model to aid doctors in their decision-making, leading to better efficiency, accuracy, and decreased workload. All leading to more growth for the business. My analysis will use the CRISP-DM (Cross-industry standard process for data mining) methodology.

Data Understanding

I am working with a dataset presented by Kaggle. The dataset was gathered from one medical center located in a prominent city in China, Guangzhou. “Chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children’s Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patients’ routine clinical care.” The dataset contains almost 6,000 images in two categories (Pneumonia/Normal). It is important to note that this is data from one location, and the images are all children, which causes some limitations. I will discuss these in more detail in the limitations section at the end of the notebook. Below is a link to the dataset.

1* Citation/Data: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia?resource=download

Data Preparation

The data needs some initial transformations/preparation to ensure the images can be used by the model. I have to change the shape and dimensions of the images and labels to do this. I also need to standardize the images to reduce their complexity, which will help improve model performance and efficiency. Additionally, class imbalance is present in the datasets, with the minority class being “normal”. Adjusting the class weights will provide more weight to the minority class and less weight to the majority class. This will help reduce bias within the model.

- Directory path for train/test/valid images
- Create image/label datasets and reshape
- Change dimensions of images and labels
- Standardize images
- Class Imbalance - class weights

In [1]:

```
#My environment required me to install certain packages and versions. Plea
#If you need to do so then please uncomment the below code.

#!pip install scikeras
#!pip install scikit-learn==1.2.2
#!pip install lime
#!pip install scikit-image
```

In [2]:

```
#Import the necessary libraries for image manipulation, graphing, preproce
#that they were initiated in the notebook.
import numpy as np
import os

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.utils import array_to_img, img_to_array, load_img

from sklearn.utils.class_weight import compute_class_weight

import tensorflow as tf
from tensorflow import keras
from keras import models
from keras import layers

from scikeras.wrappers import KerasClassifier
from sklearn.model_selection import GridSearchCV

from keras.callbacks import EarlyStopping

import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

from IPython import get_ipython
from IPython.display import display

import lime
```

```
from lime import lime_image
from skimage.io import imread
from skimage.segmentation import mark_boundaries
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import preprocess_input
```

In [3]:

```
#load the dataset in from kaggle, uncomment the below code to download the

#!kaggle datasets download -d paultimothymooney/chest-xray-pneumonia
```

In [4]:

```
#unzip the dataset so it can be loaded into the notebook. Uncomment the be
#most of the notebook.

#!unzip chest-xray-pneumonia.zip
```

In [5]:

```
# Directory path
train_data_dir = 'chest_xray/train'
test_data_dir = 'chest_xray/test'
valid_data_dir = 'chest_xray/val'
```

In [6]:

```
# Get all the data from the directory, and reshape them to put into their
def get_reshape_images(dir, bs):
  '''
  The function will take in the directory path (dir) for the image as well
  and the labels to (labels).
  '''
  generator = ImageDataGenerator().flow_from_directory(
          dir, target_size=(128, 128), batch_size=bs)

  images, labels = next(generator)
  return images, labels
```

In [7]:

```
#Using our function to reshape images and assign the train images and labe
train_images, train_labels = get_reshape_images(train_data_dir, 5216)
```

```
Found 5216 images belonging to 2 classes.
```

In [8]:

```
#Using our function to reshape images and assign the test images and label
test_images, test_labels = get_reshape_images(test_data_dir, 624)
```

```
Found 624 images belonging to 2 classes.
```

In [9]:

```
#Using our function to reshape images and assign the valid images and labe
valid_images, valid_labels = get_reshape_images(valid_data_dir, 16)
```

```
Found 16 images belonging to 2 classes.
```

In [10]:

```
 #Need to manipulate the dimensions of the images and labels so they can b
#(5216,1) for the training dataset.
 def reshape_input_image_label(images, m, labels):
  '''
  Will take in the image dataset (images) and reshape them to the desired
  It will also take in the label dataset (labels) and reshape them to the
```

```
    It will also take in the label dataset (labels) and reshape them to the
    '''
    img_unrow = images.reshape(m, -1)
    labels_final = labels.T[[1]].T
    return img_unrow, labels_final
```

In [11]:

```
#Using our function to change dimensions of the train images and labels, w
train_img_unrow, train_label_final = reshape_input_image_label(train_image
```

In [12]:

```
#Using our function to change dimensions of the test images and labels, wi
test_img_unrow, test_label_final = reshape_input_image_label(test_images,
```

In [13]:

```
#Using our function to change dimensions of the valid images and labels, w
valid_img_unrow, valid_label_final = reshape_input_image_label(valid_image
```

In [14]:

```
#I want to standardize the images for better efficiency and results. Helps
#between 0 and 1.
def standardize(img_unrow):
    '''
    Takes in the image dataset (img_unrow) and standardizes it.
    '''
    img_final = img_unrow/255
    return img_final
```

In [15]:

```
#Standardize the train images.
train_img_final = standardize(train_img_unrow)
```

In [16]:

```
#Standardize the test images.
test_img_final= standardize(test_img_unrow)
```

In [17]:

```
#Standardize the valid images.
valid_img_final = standardize(valid_img_unrow)
```

In [18]:

```
#Currently, the dataset has class imbalance in favor of pneumonia. To fix
class_weights_train = compute_class_weight(class_weight='balanced',
                                           classes=np.unique(train_label_final)
                                           y=train_label_final.ravel())

#Convert to dictionary format so it can be passed into the model.fit.
class_weights_dict_train = {0: class_weights_train[0], 1: class_weights_t

#2*Citation
```

Model

I am trying to assess two categories within an X-ray image, so I will use a binary image classification neural network. My goal to ensure my model performs the same on the training images as on unseen images. I will use the three different datasets to ensure training images as on unseen images. I will use the three different datasets to ensure the model is generalizable (train/test/valid). Since I am focused on minimizing the false negatives, I will be looking at recall as my metric for these models.

Baseline

Modeling is an iterative process, so a baseline model is needed. I will use the baseline model as a reference point as I try to improve the performance of my model. I will create a simple model to represent this. The model will only contain two neurons in its initial layer. I chose an epoch of 6 and batch size of 3 because this combination ensures all images will be passed through the model when training the baseline. Because this is a binary image classification problem, the output layer must contain just one neuron, and the activation function must be "sigmoid." When fit is applied, you see the use of class weights to handle class imbalance. Additionally, you will see the model takes in the training data and uses the test data for comparison, helping better understand how generalizable the model is. The train and test data showed similar loss metrics, but the recall score is scattered. To fix this issue, I will apply Gridsearch for the next model. The baseline model is currently in a function so it can be wrapped in order to be applied to Gridsearch.

Gridsearch/tuned model

I am applying Gridsearch to find the optimal parameters for my model. I will examine the number of neurons, optimizer, activation function, epochs, and batch size. After receiving the best parameters, I applied them to the "tuned model." The tuned model had much better and consistent recall values, showing improvement in the model overall. However, the loss is less for the training set than for the test set, which is a sign of overfitting. To fix overfitting, I must apply a regularization technique.

Regularization

For this model, I applied the Dropout technique. This randomly selects neurons to use and removes the others (remember, my model currently has 20 neurons). This will reduce some of the noise the model is picking up on, which is causing it to overfit. The results for both the recall and loss scores show a decrease in overfitting from the tuned model. There is still a bit of overfitting, but if I continue to try and regularize the model, the metrics will start to fluctuate greatly. Given the limitations of the dataset, this is the best-performing model at this time. I will give a visual of the performance of the regularized model on the test images/labels via a confusion matrix.

Side note:

The models and gridsearch can take awhile to run. If you plan on experimenting with

The models and gridsearch can take awhile to run. If you plan on experimenting with the models/results then I suggest saving after you run the models or gridsearch so you do not have to wait for results everytime. Here is some code from google colab to do so:

Calling `save('my_model.h5')` creates a h5 file `my_model.h5`.

model.save("my_h5_model.h5")

Load model back in.

reconstructed_model = keras.models.load_model("my_h5_model.h5")

Baseline Model

In [19]:

```
#Building baseline model. Passing into function so it can be used in grid
def build_baseline(optimizer='SGD', activation='relu',neurons=2):
  baseline_model = models.Sequential()
  baseline_model.add(layers.Dense(neurons, activation=activation, input_s
  baseline_model.add(layers.Dense(1, activation='sigmoid'))
  baseline_model.compile(optimizer=optimizer,
                         loss='binary_crossentropy',
                         metrics=['recall'])
  return baseline_model
```

In [20]:

```
#Fitting baseline model to see the results.
baseline_model = build_baseline()
baseline_model.fit(train_img_final,
                   train_label_final,
                   epochs=6,
                   batch_size=3,
                   class_weight = class_weights_dict_train,
                   validation_data=(test_img_final, test_label_final))
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer.
When using Sequential models, prefer using an `Input(shape)` object as the
first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/6
1739/1739 ━━━━━━━━━━━━━━━━━━━━ 8s 4ms/step - loss: 0.6860 - recall: 0.8611
- val_loss: 0.6939 - val_recall: 0.0000e+00
Epoch 2/6
1739/1739 ━━━━━━━━━━━━━━━━━━━━ 8s 3ms/step - loss: 0.6924 - recall: 0.4611
- val_loss: 0.6962 - val_recall: 0.0000e+00
Epoch 3/6
1739/1739 ━━━━━━━━━━━━━━━━━━━━ 11s 4ms/step - loss: 0.6923 - recall: 0.3771
- val_loss: 0.6964 - val_recall: 0.0000e+00
Epoch 4/6
1739/1739 ━━━━━━━━━━━━━━━━━━━━ 13s 5ms/step - loss: 0.6940 - recall: 0.3292
- val_loss: 0.6951 - val_recall: 0.0000e+00
Epoch 5/6
1739/1739 ━━━━━━━━━━━━━━━━━━━━ 6s 3ms/step - loss: 0.6964 - recall: 0.2580
- val_loss: 0.6956 - val_recall: 0.0000e+00
Epoch 6/6
Epoch 6/6
1739/1739 ━━━━━━━━━━━━━━━━━━━━ 12s 4ms/step - loss: 0.6844 - recall: 0.6687
- val_loss: 0.7032 - val_recall: 0.0000e+00
```

Out[20]:

```
<keras.src.callbacks.history.History at 0x795646296190>
```

Gridsearch/Tuned Model

In [21]:

```
#Apply early stoppage so the model reduces on the time spent running each
early_stopping = [EarlyStopping(monitor='val_loss', patience=5)]

#3*Citation
```

In [22]:

```
#Wrap the model in KerasClassifier so it can be used for the gridsearch.
tuning_model=KerasClassifier(build_fn=build_baseline)
```

In [23]:

```
#Create the grid to see what parameters work best for the model. The grids
#best parameters so if you run the gridsearch it will save you some time.
param_grid = {
    "model__neurons": [20], #[10,20,30]
    "model__optimizer": ['SGD','adam'], #['adam', 'RMSProp']
    "model__activation" : ['sigmoid'], #['tanh', 'relu'],
    "epochs": [50], #[40,50,60]
    "batch_size" : [20] #[10,20,30,40]
}
```

In [24]:

```
#Set up the grid search on the model and apply a cross validation of 3 fo
gs=GridSearchCV(estimator=tuning_model, param_grid=param_grid, cv=3)
gs = gs.fit(test_img_final, test_label_final, class_weight = class_weights

#4*Citation
```

```
/usr/local/lib/python3.11/dist-packages/scikeras/wrappers.py:925: UserWarni
ng: ``build_fn`` will be renamed to ``model`` in a future release, at which
point use of ``build_fn`` will raise an Error instead.
  X, y = self._initialize(X, y)
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer.
When using Sequential models, prefer using an `Input(shape)` object as the
first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 2s 22ms/step - loss: 0.8194 - recall: 0.2060
Epoch 2/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7298 - recall: 0.0455
Epoch 3/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.6904 - recall: 0.2588
Epoch 4/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.6750 - recall: 0.4813
Epoch 5/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - loss: 0.6012 - recall: 0.4484
Epoch 6/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - loss: 0.5220 - recall: 0.6809
Epoch 7/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 35ms/step - loss: 0.5462 - recall: 0.5366
Epoch 8/50
```

```
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 32ms/step - loss: 0.4978 - recall: 0.6297
Epoch 9/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - loss: 0.5233 - recall: 0.6435
Epoch 10/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 0.4701 - recall: 0.7015
Epoch 11/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.4404 - recall: 0.6904
Epoch 12/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.4544 - recall: 0.7472
Epoch 13/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.4261 - recall: 0.6981
Epoch 14/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.4136 - recall: 0.7753
Epoch 15/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.4493 - recall: 0.7351
Epoch 16/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.3942 - recall: 0.6816
Epoch 17/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.4078 - recall: 0.7430
Epoch 18/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.4395 - recall: 0.6911
Epoch 19/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.3321 - recall: 0.8167
Epoch 20/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.3959 - recall: 0.7822
Epoch 21/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.3370 - recall: 0.7876
Epoch 22/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.4877 - recall: 0.7020
Epoch 23/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 0.3211 - recall: 0.8079
Epoch 24/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 0.3412 - recall: 0.7543
Epoch 25/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - loss: 0.3184 - recall: 0.8232
Epoch 26/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - loss: 0.3633 - recall: 0.7690
Epoch 27/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - loss: 0.3273 - recall: 0.7703
Epoch 28/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 33ms/step - loss: 0.3623 - recall: 0.8237
Epoch 29/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - loss: 0.3240 - recall: 0.7835
Epoch 30/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - loss: 0.3456 - recall: 0.7911
Epoch 31/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 23ms/step - loss: 0.3844 - recall: 0.7533
Epoch 32/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 0.2804 - recall: 0.8269
Epoch 33/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 0.2937 - recall: 0.8323
Epoch 34/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.2788 - recall: 0.8247
Epoch 35/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 23ms/step - loss: 0.2764 - recall: 0.8372
Epoch 36/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - loss: 0.3637 - recall: 0.7983
Epoch 37/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.2671 - recall: 0.8578
Epoch 38/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 0.3226 - recall: 0.7953
```

```
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 0.3226 - recall: 0.7953
Epoch 39/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.3332 - recall: 0.7575
Epoch 40/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.3209 - recall: 0.8084
Epoch 41/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - loss: 0.3208 - recall: 0.8158
Epoch 42/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.2715 - recall: 0.8175
Epoch 43/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step - loss: 0.2629 - recall: 0.8422
Epoch 44/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 0.2490 - recall: 0.8756
Epoch 45/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.2616 - recall: 0.8556
Epoch 46/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 0.2229 - recall: 0.8586
Epoch 47/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.3235 - recall: 0.7918
Epoch 48/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 31ms/step - loss: 0.2459 - recall: 0.8400
Epoch 49/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 23ms/step - loss: 0.2581 - recall: 0.8556
Epoch 50/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 25ms/step - loss: 0.2469 - recall: 0.8648
11/11 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step
/usr/local/lib/python3.11/dist-packages/scikeras/wrappers.py:925: UserWarni
ng: ``build_fn`` will be renamed to ``model`` in a future release, at which
point use of ``build_fn`` will raise an Error instead.
  X, y = self._initialize(X, y)
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer.
When using Sequential models, prefer using an `Input(shape)` object as the
first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 3s 10ms/step - loss: 0.8973 - recall: 0.0000e+00
Epoch 2/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.7471 - recall: 0.0000e+00
Epoch 3/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.7062 - recall: 0.0000e+00
Epoch 4/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.7003 - recall: 0.0037
Epoch 5/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.6397 - recall: 0.1711
Epoch 6/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - loss: 0.6398 - recall: 0.2969
Epoch 7/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 0.5577 - recall: 0.5838
Epoch 8/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.5172 - recall: 0.6493
Epoch 9/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.4833 - recall: 0.6757
Epoch 10/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.4872 - recall: 0.6542
Epoch 11/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.4696 - recall: 0.6869
Epoch 12/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.4797 - recall: 0.7124
Epoch 13/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.4032 - recall: 0.7891
```

```
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.4032 - recall: 0.7891
Epoch 14/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.4667 - recall: 0.7275
Epoch 15/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.4095 - recall: 0.7873
Epoch 16/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.4092 - recall: 0.7620
Epoch 17/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.3998 - recall: 0.8037
Epoch 18/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3944 - recall: 0.7604
Epoch 19/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.3884 - recall: 0.8141
Epoch 20/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.3789 - recall: 0.7464
Epoch 21/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.3916 - recall: 0.7285
Epoch 22/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.3626 - recall: 0.7850
Epoch 23/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3513 - recall: 0.8468
Epoch 24/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.3411 - recall: 0.8047
Epoch 25/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3733 - recall: 0.7906
Epoch 26/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.3184 - recall: 0.8331
Epoch 27/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.3811 - recall: 0.7301
Epoch 28/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3333 - recall: 0.8507
Epoch 29/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.3352 - recall: 0.8180
Epoch 30/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.3229 - recall: 0.8363
Epoch 31/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3281 - recall: 0.8088
Epoch 32/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.4439 - recall: 0.7758
Epoch 33/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - loss: 0.3365 - recall: 0.8489
Epoch 34/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - loss: 0.3103 - recall: 0.8908
Epoch 35/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - loss: 0.2901 - recall: 0.8489
Epoch 36/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - loss: 0.2795 - recall: 0.8702
Epoch 37/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - loss: 0.2880 - recall: 0.8523
Epoch 38/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - loss: 0.3549 - recall: 0.7934
Epoch 39/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 0.2646 - recall: 0.8812
Epoch 40/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2514 - recall: 0.8760
Epoch 41/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.2959 - recall: 0.8491
Epoch 42/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.3116 - recall: 0.8140
Epoch 43/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.3642 - recall: 0.7875
```

```
Epoch 44/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.2948 - recall: 0.8531
Epoch 45/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.2694 - recall: 0.8528
Epoch 46/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2917 - recall: 0.8796
Epoch 47/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.2691 - recall: 0.8788
Epoch 48/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.2901 - recall: 0.8476
Epoch 49/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.2697 - recall: 0.8690
Epoch 50/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.3547 - recall: 0.7777
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step
/usr/local/lib/python3.11/dist-packages/scikeras/wrappers.py:925: UserWarni
ng: ``build_fn`` will be renamed to ``model`` in a future release, at which
point use of ``build_fn`` will raise an Error instead.
  X, y = self._initialize(X, y)
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer.
When using Sequential models, prefer using an `Input(shape)` object as the
first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 13ms/step - loss: 0.8128 - recall: 0.1362
Epoch 2/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.7682 - recall: 0.2719
Epoch 3/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.6913 - recall: 0.3454
Epoch 4/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.6806 - recall: 0.3499
Epoch 5/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - loss: 0.5640 - recall: 0.5946
Epoch 6/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.5610 - recall: 0.6734
Epoch 7/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.5385 - recall: 0.6263
Epoch 8/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.5223 - recall: 0.6781
Epoch 9/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - loss: 0.5017 - recall: 0.6709
Epoch 10/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.4732 - recall: 0.6648
Epoch 11/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.4369 - recall: 0.7308
Epoch 12/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - loss: 0.4487 - recall: 0.6733
Epoch 13/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.4199 - recall: 0.7890
Epoch 14/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.5045 - recall: 0.5957
Epoch 15/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.4037 - recall: 0.7509
Epoch 16/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3667 - recall: 0.8168
Epoch 17/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 0.4688 - recall: 0.6984
Epoch 18/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - loss: 0.3681 - recall: 0.7779
```

```
Epoch 19/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - loss: 0.4153 - recall: 0.6934
Epoch 20/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.3892 - recall: 0.7469
Epoch 21/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - loss: 0.3322 - recall: 0.8319
Epoch 22/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.3647 - recall: 0.8469
Epoch 23/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - loss: 0.3717 - recall: 0.7481
Epoch 24/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - loss: 0.3662 - recall: 0.7375
Epoch 25/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - loss: 0.3656 - recall: 0.7691
Epoch 26/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.3864 - recall: 0.7834
Epoch 27/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.2696 - recall: 0.8753
Epoch 28/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.2988 - recall: 0.8373
Epoch 29/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.4133 - recall: 0.7698
Epoch 30/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2865 - recall: 0.8455
Epoch 31/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.3231 - recall: 0.8150
Epoch 32/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3761 - recall: 0.7209
Epoch 33/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.2831 - recall: 0.8781
Epoch 34/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3765 - recall: 0.7304
Epoch 35/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.2747 - recall: 0.8405
Epoch 36/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.2989 - recall: 0.8330
Epoch 37/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.2607 - recall: 0.8652
Epoch 38/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2669 - recall: 0.8458
Epoch 39/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.2324 - recall: 0.8756
Epoch 40/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2535 - recall: 0.8144
Epoch 41/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2786 - recall: 0.8791
Epoch 42/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3223 - recall: 0.7822
Epoch 43/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2669 - recall: 0.8648
Epoch 44/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.3329 - recall: 0.8384
Epoch 45/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3440 - recall: 0.7850
Epoch 46/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2551 - recall: 0.8564
Epoch 47/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.2652 - recall: 0.8155
Epoch 48/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2972 - recall: 0.8543
Epoch 49/50
Epoch 49/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - loss: 0.2419 - recall: 0.9093
Epoch 50/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3659 - recall: 0.7458
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step
/usr/local/lib/python3.11/dist-packages/scikeras/wrappers.py:925: UserWarni
ng: ``build_fn`` will be renamed to ``model`` in a future release, at which
point use of ``build_fn`` will raise an Error instead.
  X, y = self._initialize(X, y)
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer.
When using Sequential models, prefer using an `Input(shape)` object as the
first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 2s 21ms/step - loss: 1.2215 - recall: 0.1538
Epoch 2/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 1.1384 - recall: 0.0000e+00
Epoch 3/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - loss: 1.0932 - recall: 0.0000e+00
Epoch 4/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - loss: 0.9713 - recall: 0.0000e+00
Epoch 5/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 30ms/step - loss: 0.9702 - recall: 0.0000e+00
Epoch 6/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 0.9177 - recall: 0.0000e+00
Epoch 7/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 0.8786 - recall: 0.0000e+00
Epoch 8/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.8308 - recall: 0.0000e+00
Epoch 9/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 0.8086 - recall: 0.0000e+00
Epoch 10/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.8043 - recall: 0.0000e+00
Epoch 11/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 0.7823 - recall: 0.0000e+00
Epoch 12/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 0.7697 - recall: 0.0000e+00
Epoch 13/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7620 - recall: 0.0000e+00
Epoch 14/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7607 - recall: 0.0000e+00
Epoch 15/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.7565 - recall: 0.0000e+00
Epoch 16/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7574 - recall: 0.0000e+00
Epoch 17/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step - loss: 0.7571 - recall: 0.0000e+00
Epoch 18/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 0.7599 - recall: 0.0000e+00
Epoch 19/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7582 - recall: 0.0000e+00
Epoch 20/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7621 - recall: 0.0000e+00
Epoch 21/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7607 - recall: 0.0000e+00
Epoch 22/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7493 - recall: 0.0000e+00
Epoch 23/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 26ms/step - loss: 0.7578 - recall: 0.0000e+00
Epoch 24/50
Epoch 24/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - loss: 0.7530 - recall: 0.0000e+00
Epoch 25/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - loss: 0.7568 - recall: 0.0000e+00
Epoch 26/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - loss: 0.7577 - recall: 0.0000e+00
Epoch 27/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - loss: 0.7471 - recall: 0.0000e+00
Epoch 28/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - loss: 0.7609 - recall: 0.0000e+00
Epoch 29/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.7617 - recall: 0.0000e+00
Epoch 30/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7572 - recall: 0.0000e+00
Epoch 31/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - loss: 0.7500 - recall: 0.0000e+00
Epoch 32/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 0.7501 - recall: 0.0000e+00
Epoch 33/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - loss: 0.7477 - recall: 0.0000e+00
Epoch 34/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7569 - recall: 0.0000e+00
Epoch 35/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - loss: 0.7603 - recall: 0.0000e+00
Epoch 36/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 0.7512 - recall: 0.0000e+00
Epoch 37/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - loss: 0.7536 - recall: 0.0000e+00
Epoch 38/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.7490 - recall: 0.0000e+00
Epoch 39/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7543 - recall: 0.0000e+00
Epoch 40/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 0.7506 - recall: 0.0000e+00
Epoch 41/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - loss: 0.7598 - recall: 0.0000e+00
Epoch 42/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 0.7554 - recall: 0.0000e+00
Epoch 43/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - loss: 0.7463 - recall: 0.0000e+00
Epoch 44/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 0.7551 - recall: 0.0000e+00
Epoch 45/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 0.7568 - recall: 0.0000e+00
Epoch 46/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7579 - recall: 0.0000e+00
Epoch 47/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.7526 - recall: 0.0000e+00
Epoch 48/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - loss: 0.7651 - recall: 0.0000e+00
Epoch 49/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - loss: 0.7566 - recall: 0.0000e+00
Epoch 50/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - loss: 0.7618 - recall: 0.0000e+00
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step
/usr/local/lib/python3.11/dist-packages/scikeras/wrappers.py:925: UserWarni
ng: ``build_fn`` will be renamed to ``model`` in a future release, at which
point use of ``build_fn`` will raise an Error instead.
  X, y = self._initialize(X, y)
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer.
```

```
When using Sequential models, prefer using an `Input(shape)` object as the
first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 3s 19ms/step - loss: 1.0545 - recall: 0.1583
Epoch 2/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - loss: 0.9600 - recall: 0.0000e+00
Epoch 3/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.9454 - recall: 0.0000e+00
Epoch 4/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.8677 - recall: 0.0000e+00
Epoch 5/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 0.8327 - recall: 0.0000e+00
Epoch 6/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.8025 - recall: 0.0000e+00
Epoch 7/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7913 - recall: 0.0000e+00
Epoch 8/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.7802 - recall: 0.0000e+00
Epoch 9/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 0.7707 - recall: 0.0000e+00
Epoch 10/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.7609 - recall: 0.0000e+00
Epoch 11/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7585 - recall: 0.0000e+00
Epoch 12/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7601 - recall: 0.0000e+00
Epoch 13/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7559 - recall: 0.0000e+00
Epoch 14/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7519 - recall: 0.0000e+00
Epoch 15/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 0.7549 - recall: 0.0000e+00
Epoch 16/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7522 - recall: 0.0000e+00
Epoch 17/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - loss: 0.7523 - recall: 0.0000e+00
Epoch 18/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - loss: 0.7487 - recall: 0.0000e+00
Epoch 19/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 33ms/step - loss: 0.7504 - recall: 0.0000e+00
Epoch 20/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 26ms/step - loss: 0.7582 - recall: 0.0000e+00
Epoch 21/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 31ms/step - loss: 0.7498 - recall: 0.0000e+00
Epoch 22/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.7557 - recall: 0.0000e+00
Epoch 23/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7527 - recall: 0.0000e+00
Epoch 24/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 0.7601 - recall: 0.0000e+00
Epoch 25/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7615 - recall: 0.0000e+00
Epoch 26/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 0.7618 - recall: 0.0000e+00
Epoch 27/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7534 - recall: 0.0000e+00
Epoch 28/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.7485 - recall: 0.0000e+00
Epoch 29/50
```

```
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.7533 - recall: 0.0000e+00
Epoch 30/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7601 - recall: 0.0000e+00
Epoch 31/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7495 - recall: 0.0000e+00
Epoch 32/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7618 - recall: 0.0000e+00
Epoch 33/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 0.7513 - recall: 0.0000e+00
Epoch 34/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 0.7581 - recall: 0.0000e+00
Epoch 35/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.7492 - recall: 0.0000e+00
Epoch 36/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7492 - recall: 0.0000e+00
Epoch 37/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7578 - recall: 0.0000e+00
Epoch 38/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.7539 - recall: 0.0000e+00
Epoch 39/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7560 - recall: 0.0000e+00
Epoch 40/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - loss: 0.7529 - recall: 0.0000e+00
Epoch 41/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - loss: 0.7577 - recall: 0.0000e+00
Epoch 42/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - loss: 0.7564 - recall: 0.0000e+00
Epoch 43/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - loss: 0.7589 - recall: 0.0000e+00
Epoch 44/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 30ms/step - loss: 0.7528 - recall: 0.0000e+00
Epoch 45/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7478 - recall: 0.0000e+00
Epoch 46/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7595 - recall: 0.0000e+00
Epoch 47/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 0.7571 - recall: 0.0000e+00
Epoch 48/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.7566 - recall: 0.0000e+00
Epoch 49/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7521 - recall: 0.0000e+00
Epoch 50/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step - loss: 0.7570 - recall: 0.0000e+00
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step
/usr/local/lib/python3.11/dist-packages/scikeras/wrappers.py:925: UserWarni
ng: ``build_fn`` will be renamed to ``model`` in a future release, at which
point use of ``build_fn`` will raise an Error instead.
  X, y = self._initialize(X, y)
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer.
When using Sequential models, prefer using an `Input(shape)` object as the
first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 2s 19ms/step - loss: 0.8820 - recall: 0.1816
Epoch 2/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.8755 - recall: 0.0000e+00
Epoch 3/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.8156 - recall: 0.0000e+00
Epoch 4/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.8155 - recall: 0.0000e+00
```

```
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 0.8155 - recall: 0.0000e+00
Epoch 5/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 0.7647 - recall: 0.0000e+00
Epoch 6/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 0.7676 - recall: 0.0000e+00
Epoch 7/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 0.7628 - recall: 0.0000e+00
Epoch 8/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - loss: 0.7578 - recall: 0.0000e+00
Epoch 9/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - loss: 0.7580 - recall: 0.0000e+00
Epoch 10/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - loss: 0.7556 - recall: 0.0000e+00
Epoch 11/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - loss: 0.7556 - recall: 0.0000e+00
Epoch 12/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - loss: 0.7569 - recall: 0.0000e+00
Epoch 13/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 25ms/step - loss: 0.7567 - recall: 0.0000e+00
Epoch 14/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.7562 - recall: 0.0000e+00
Epoch 15/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.7512 - recall: 0.0000e+00
Epoch 16/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7542 - recall: 0.0000e+00
Epoch 17/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.7564 - recall: 0.0000e+00
Epoch 18/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 0.7556 - recall: 0.0000e+00
Epoch 19/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 0.7527 - recall: 0.0000e+00
Epoch 20/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7571 - recall: 0.0000e+00
Epoch 21/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 0.7534 - recall: 0.0000e+00
Epoch 22/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7531 - recall: 0.0000e+00
Epoch 23/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7481 - recall: 0.0000e+00
Epoch 24/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7534 - recall: 0.0000e+00
Epoch 25/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7566 - recall: 0.0000e+00
Epoch 26/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7490 - recall: 0.0000e+00
Epoch 27/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7599 - recall: 0.0000e+00
Epoch 28/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 0.7554 - recall: 0.0000e+00
Epoch 29/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.7590 - recall: 0.0000e+00
Epoch 30/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7567 - recall: 0.0000e+00
Epoch 31/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 25ms/step - loss: 0.7530 - recall: 0.0000e+00
Epoch 32/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 29ms/step - loss: 0.7606 - recall: 0.0000e+00
Epoch 33/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 2s 63ms/step - loss: 0.7558 - recall: 0.0000e+00
Epoch 34/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 2s 28ms/step - loss: 0.7622 - recall: 0.0000e+00
```

```
Epoch 35/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - loss: 0.7538 - recall: 0.0000e+00
Epoch 36/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 26ms/step - loss: 0.7516 - recall: 0.0000e+00
Epoch 37/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 0.7604 - recall: 0.0000e+00
Epoch 38/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7614 - recall: 0.0000e+00
Epoch 39/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - loss: 0.7549 - recall: 0.0000e+00
Epoch 40/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step - loss: 0.7595 - recall: 0.0000e+00
Epoch 41/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7567 - recall: 0.0000e+00
Epoch 42/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7555 - recall: 0.0000e+00
Epoch 43/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7643 - recall: 0.0000e+00
Epoch 44/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7557 - recall: 0.0000e+00
Epoch 45/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7523 - recall: 0.0000e+00
Epoch 46/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 0.7523 - recall: 0.0000e+00
Epoch 47/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.7535 - recall: 0.0000e+00
Epoch 48/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step - loss: 0.7541 - recall: 0.0000e+00
Epoch 49/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.7438 - recall: 0.0000e+00
Epoch 50/50
21/21 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - loss: 0.7644 - recall: 0.0000e+00
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step
/usr/local/lib/python3.11/dist-packages/scikeras/wrappers.py:925: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, at which point use of ``build_fn`` will raise an Error instead.
  X, y = self._initialize(X, y)
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 0.7389 - recall: 0.2059
Epoch 2/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.7371 - recall: 0.1716
Epoch 3/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.6683 - recall: 0.5063
Epoch 4/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.6403 - recall: 0.3360
Epoch 5/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.5076 - recall: 0.6627
Epoch 6/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - loss: 0.4947 - recall: 0.6753
Epoch 7/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.5011 - recall: 0.6330
Epoch 8/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - loss: 0.4390 - recall: 0.7745
Epoch 9/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.7346 - recall: 0.1199
```

```
Epoch 10/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 0.4644 - recall: 0.7346
Epoch 11/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 0.4586 - recall: 0.7406
Epoch 12/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.4256 - recall: 0.7364
Epoch 13/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 0.4349 - recall: 0.7677
Epoch 14/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 0.3709 - recall: 0.7974
Epoch 15/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 0.4511 - recall: 0.6486
Epoch 16/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.3503 - recall: 0.7894
Epoch 17/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3473 - recall: 0.7750
Epoch 18/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 0.3835 - recall: 0.7165
Epoch 19/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 0.4355 - recall: 0.7457
Epoch 20/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 0.3433 - recall: 0.8226
Epoch 21/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 0.3652 - recall: 0.7544
Epoch 22/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 0.3193 - recall: 0.8290
Epoch 23/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 0.3253 - recall: 0.8118
Epoch 24/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3259 - recall: 0.8219
Epoch 25/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2846 - recall: 0.8228
Epoch 26/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2955 - recall: 0.8242
Epoch 27/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - loss: 0.3047 - recall: 0.8386
Epoch 28/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - loss: 0.3584 - recall: 0.7540
Epoch 29/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - loss: 0.3749 - recall: 0.7112
Epoch 30/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.2950 - recall: 0.8021
Epoch 31/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 0.3313 - recall: 0.8279
Epoch 32/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.3329 - recall: 0.8143
Epoch 33/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 0.3217 - recall: 0.8009
Epoch 34/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 0.4273 - recall: 0.8038
Epoch 35/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.3295 - recall: 0.7780
Epoch 36/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 0.2848 - recall: 0.8481
Epoch 37/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 0.2815 - recall: 0.8384
Epoch 38/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 0.2811 - recall: 0.8505
Epoch 39/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - loss: 0.2868 - recall: 0.7988
Epoch 40/50
Epoch 40/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2855 - recall: 0.8145
Epoch 41/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.2781 - recall: 0.8305
Epoch 42/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 0.2927 - recall: 0.8162
Epoch 43/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 13ms/step - loss: 0.2224 - recall: 0.8829
Epoch 44/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 0.2248 - recall: 0.8503
Epoch 45/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 0.2607 - recall: 0.8321
Epoch 46/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 0.2849 - recall: 0.8215
Epoch 47/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2673 - recall: 0.8496
Epoch 48/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 0.2606 - recall: 0.8588
Epoch 49/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2529 - recall: 0.8712
Epoch 50/50
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - loss: 0.2544 - recall: 0.8429
```

In [25]:

```
#See the results of the gridsearch.
best_params=gs.best_params_
best_params
```

Out[25]:

```
{'batch_size': 20,
 'epochs': 50,
 'model__activation': 'sigmoid',
 'model__neurons': 20,
 'model__optimizer': 'SGD'}
```

In [26]:

```
#Applying the parameters given by the gridsearch and see the model result
tuned_model = models.Sequential()
tuned_model.add(layers.Dense(20, activation='sigmoid', input_shape=(49152
tuned_model.add(layers.Dense(1, activation='sigmoid'))
tuned_model.compile(optimizer='SGD',
                      loss='binary_crossentropy',
                      metrics=['recall'])
tuned_model.fit(train_img_final,
                    train_label_final,
                    epochs=50,
                    batch_size=20,
                    callbacks=early_stopping,
                    class_weight = class_weights_dict_train,
                    validation_data=(test_img_final, test_label_final))
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer.
When using Sequential models, prefer using an `Input(shape)` object as the
first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 5s 16ms/step - loss: 0.5909 - recall: 0.6728 -
val_loss: 0.4031 - val_recall: 0.8769
Epoch 2/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 4s 12ms/step - loss: 0.3227 - recall: 0.8731 -
val_loss: 0.3900 - val_recall: 0.9487
```

```
val_loss: 0.5960 - val_recall: 0.9487
Epoch 3/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 3s 12ms/step - loss: 0.2486 - recall: 0.9013 -
val_loss: 0.5302 - val_recall: 0.9872
Epoch 4/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 6s 17ms/step - loss: 0.2185 - recall: 0.9136 -
val_loss: 0.4828 - val_recall: 0.9667
Epoch 5/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 4s 12ms/step - loss: 0.2146 - recall: 0.9155 -
val_loss: 0.3874 - val_recall: 0.9179
Epoch 6/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 5s 12ms/step - loss: 0.1792 - recall: 0.9276 -
val_loss: 0.5328 - val_recall: 0.9769
Epoch 7/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 6s 17ms/step - loss: 0.1636 - recall: 0.9301 -
val_loss: 0.6183 - val_recall: 0.9821
Epoch 8/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 4s 12ms/step - loss: 0.1805 - recall: 0.9276 -
val_loss: 0.4652 - val_recall: 0.9308
Epoch 9/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 5s 12ms/step - loss: 0.1593 - recall: 0.9351 -
val_loss: 0.4761 - val_recall: 0.9462
Epoch 10/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 6s 15ms/step - loss: 0.1574 - recall: 0.9352 -
val_loss: 0.6513 - val_recall: 0.9821
```

Out[26]:

```
<keras.src.callbacks.history.History at 0x79563e79d190>
```

Regularization

In [27]:

```
#Applying regularization to the model due to overfitting. Will use the dr
reg_model = models.Sequential()

reg_model.add(layers.Dense(20, activation='sigmoid', input_shape=(49152,)
reg_model.add(layers.Dropout(0.5))
reg_model.add(layers.Dense(1, activation='sigmoid'))

reg_model.compile(optimizer='SGD',
                      loss='binary_crossentropy',
                      metrics=['recall'])
reg_model.fit(train_img_final,
                    train_label_final,
                    epochs=50,
                    batch_size=20,
                    callbacks=early_stopping,
                    class_weight = class_weights_dict_train,
                    validation_data=(test_img_final, test_label_final))
```

```
Epoch 1/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 5s 15ms/step - loss: 0.6956 - recall: 0.5344 -
val_loss: 0.4903 - val_recall: 0.9846
Epoch 2/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 3s 12ms/step - loss: 0.4579 - recall: 0.8225 -
val_loss: 0.4245 - val_recall: 0.9590
Epoch 3/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 6s 16ms/step - loss: 0.3561 - recall: 0.8692 -
val_loss: 0.3789 - val_recall: 0.8385
Epoch 4/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 5s 19ms/step - loss: 0.3013 - recall: 0.8948 -
val_loss: 0.4311 - val_recall: 0.9641
```

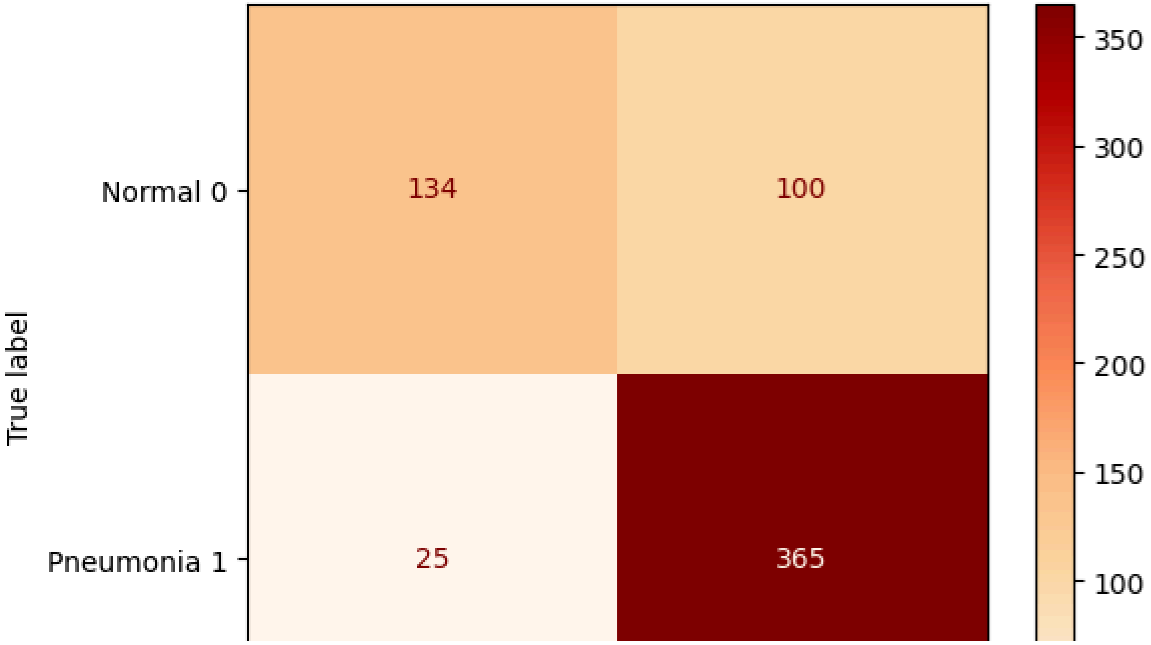
```
Epoch 5/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 3s 12ms/step - loss: 0.2702 - recall: 0.9016 -
val_loss: 0.3770 - val_recall: 0.8641
Epoch 6/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 3s 13ms/step - loss: 0.2501 - recall: 0.9124 -
val_loss: 0.3732 - val_recall: 0.9205
Epoch 7/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 3s 13ms/step - loss: 0.2514 - recall: 0.8975 -
val_loss: 0.4630 - val_recall: 0.9513
Epoch 8/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 5s 21ms/step - loss: 0.2263 - recall: 0.9123 -
val_loss: 0.3933 - val_recall: 0.9282
Epoch 9/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 3s 13ms/step - loss: 0.2102 - recall: 0.9191 -
val_loss: 0.7136 - val_recall: 0.9872
Epoch 10/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 5s 14ms/step - loss: 0.2058 - recall: 0.9264 -
val_loss: 0.6139 - val_recall: 0.9846
Epoch 11/50
261/261 ━━━━━━━━━━━━━━━━━━━━ 7s 21ms/step - loss: 0.2036 - recall: 0.9169 -
val_loss: 0.4748 - val_recall: 0.9359
```

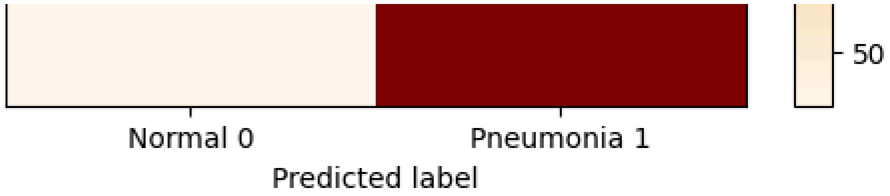
```
Out[27]: <keras.src.callbacks.history.History at 0x7956440f5190>
```

```
In [28]: # Evaluate the reg_model to see the results with the test data. Calculate
         reg_model.evaluate(test_img_final, test_label_final)

         y_pred_binary = (reg_model.predict(test_img_final) > 0.5).astype(int)
         cnf_matrix = confusion_matrix(test_label_final, y_pred_binary)

         display_labels = ['Normal 0', 'Pneumonia 1']

         disp = ConfusionMatrixDisplay(confusion_matrix=cnf_matrix, display_labels=
         disp.plot(cmap='OrRd')
```

```
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step - loss: 0.4783 - recall: 0.9340
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step
```

```
Out[28]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x79563
         e0672d0>
```





Evaluation

Finally, I must evaluate our model on the unseen data (valid images/labels). The model performed well on the unseen images, confirming our model is generalizable. I have a perfect recall score with a loss of 0.19. I used a confusion matrix to visualize the predictions. As more data is acquired, the recall score will lower slightly.

Overall, I created a baseline model as a reference point. Then I used a gridsearch to find the optimal parameters. The parameters I investigated were the epochs, batch size, activation function, optimizer, and number of neurons. After using the parameters suggested by the gridsearch, I saw the recall and loss scores demonstrated overfitting. To fix this issue, I applied regularization to the model via the dropout techinique. The model was able to reduce the amount of overfitting overall, but still had some slight overfitting. I experimented with additional layers and regularization techniques but all of those lead to diminishing results. Overall, I was left with a model containing two layers with the following parameters:

- Epochs - 50
- Batch Size - 20
- Neurons - 20
- Activation Function - sigmoid
- Optimizer - SGD

The output layer must have one nueron and sigmoid as the activation function since this is a binary image classification.

Additionally, I used LIME to take individual instances to visually show what the model was looking at in order to make its prediction. I represent two images, one with pneumonia and one without pneumonia.

Final Model

In [29]:

```
final_model = reg_model
```

In [30]:

```
#Set our final model and evaluate on the valid data.
final_model.evaluate(valid_img_final, valid_label_final)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - loss: 0.1903 - recall: 1.0000
```

Out[30]:

```
[0.19032908976078033, 1.0]
```

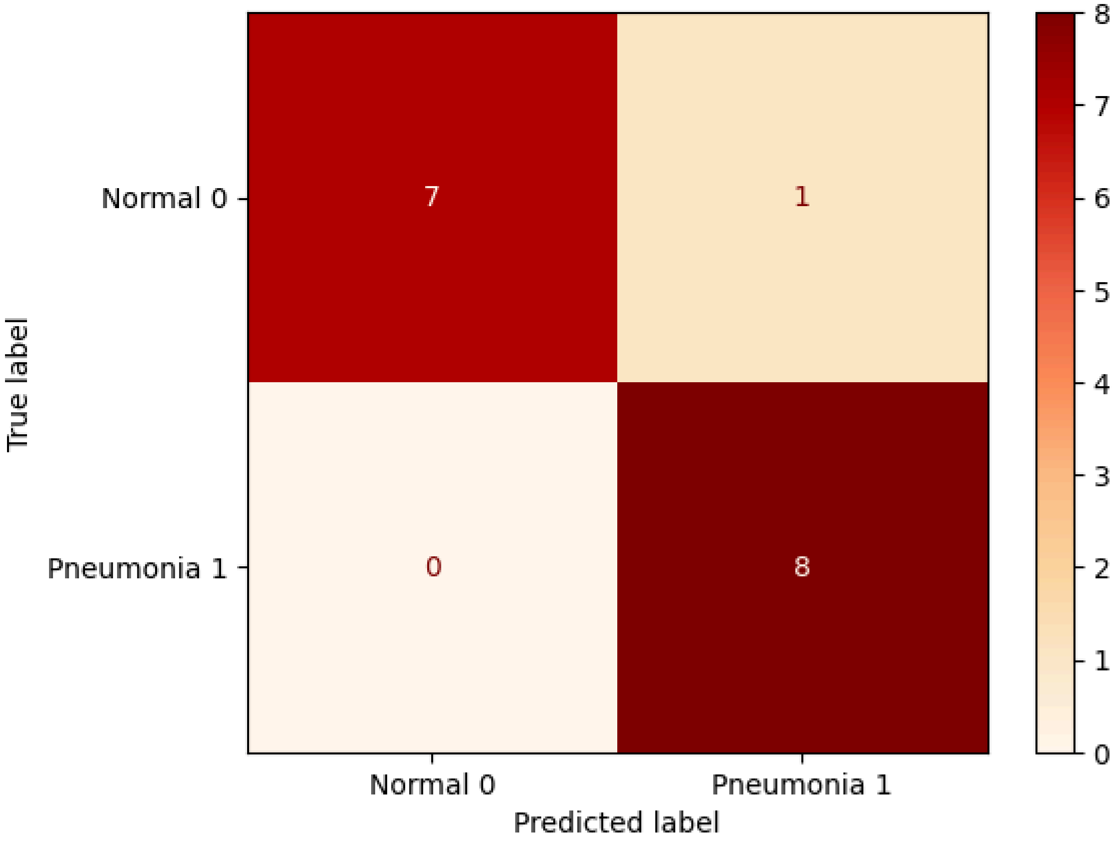
In [31]:

```
#Calculate and plot the confusion matrix to see a visualization of the re
y_pred_binary = (final_model.predict(valid_img_final) > 0.5).astype(int)
cnf_matrix = confusion_matrix(valid_label_final, y_pred_binary)

display_labels = ['Normal 0', 'Pneumonia 1']

disp = ConfusionMatrixDisplay(confusion_matrix=cnf_matrix, display_labels=
disp.plot(cmap='OrRd')
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
```

Out[31]:

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x79564
66cb750>
```



LIME Visualization

In [32]:

```
#Creat path for an image of pneumonia to be used with LIME.
img_path_pneumonia = '/content/chest_xray/val/PNEUMONIA/person1952_bacter
```

In [33]:

```
#Creat path for an image of normal to be used with LIME.
img_path_normal = '/content/chest_xray/val/NORMAL/NORMAL2-IM-1438-0001.jp
```

In [34]:

```
#Currently, our final model was trained on a flattened image dataset (Non
#So I need to reshape my images to not throw an error when running LIME e
def predict_fn(images):
  '''
  Takes in the image (images) and reshapes them. Then uses our model to m
  '''
  reshaped_images = images.reshape(images.shape[0], -1)
  predictions = final_model.predict(reshaped_images)
```

```
    predictions = final_model.predict(reshaped_images)
    return predictions
```

In [35]:

```
#Creating a function that can be reused for a normal image and a pnuemonia
#into the LIME explainer and our final model. Then it will output the orig
def show_hot_spots(imge):
    '''
    Takes in the image (imge) and reshapes/stadardizes it. Then creates an i
    our predicitions. Finally, takes the results of the explainer and create
    '''
#Load and preprocess the image
    img = image.load_img(imge, target_size=(128, 128))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    x = x / 255.0

#Create a LimeImageExplainer
    explainer = lime_image.LimeImageExplainer()

#Explain the prediction
    explanation = explainer.explain_instance(x[0],
                                             predict_fn,
                                             top_labels=2,
                                             hide_color=0,
                                             num_samples=1000)

#Finally, takes in the results from explainer and shows the positive and n
    temp, mask = explanation.get_image_and_mask(explanation.top_labels[0], p
    plt.imshow(mark_boundaries(temp / 2 + 0.5, mask))
    plt.title('LIME Explanation')

    return plt.show()

#5* Citation
#6* Citation
```

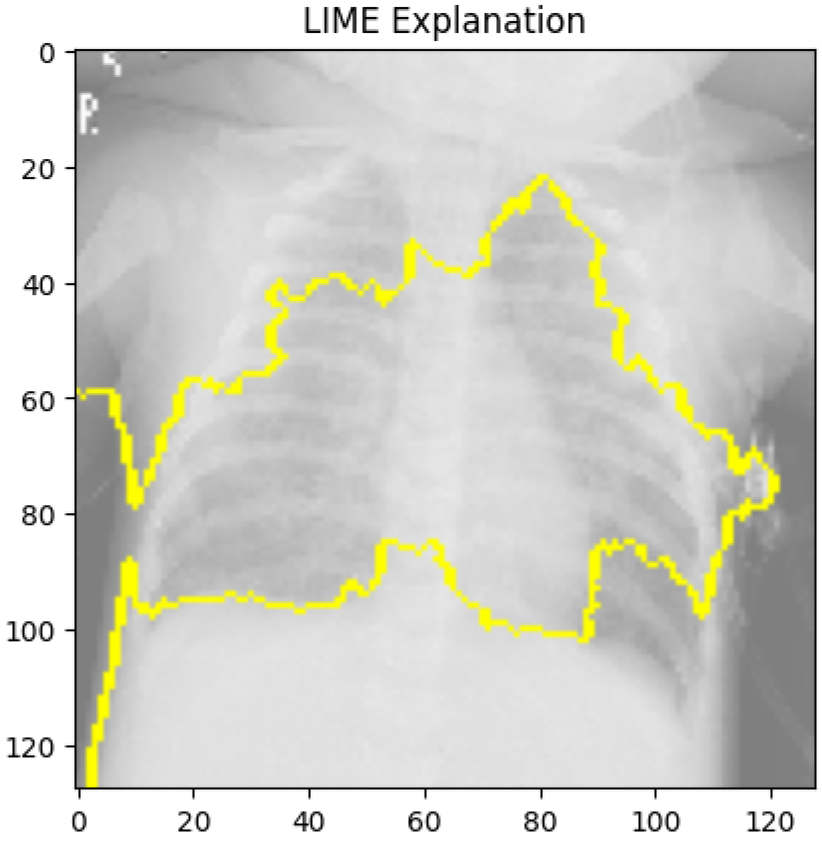
In [36]:

```
#Apply the function with an image where someone had pneumonia.
show_hot_spots(img_path_pneumonia)
```

```
  0%|          | 0/1000 [00:00<?, ?it/s]
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
```



In [37]:

```
#Apply the function with an image where someone was normal.
show_hot_spots(img_path_normal)
```

```
  0%|          | 0/1000 [00:00<?, ?it/s]
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
```