Here you can find the necessary import

In [ ]:

```
import tensorflow as tf
import os
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve, auc, precision_recall_curve, average_precision_sco
re, confusion_matrix
import pandas as pd
import matplotlib.pyplot as plt

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, BatchNormalization, Flatten, D
ense, Input
from tensorflow.keras.preprocessing.image import ImageDataGenerator

import seaborn as sns
```

```
WARNING:tensorflow:From C:\Users\Faiza Anan Noor\anaconda3\Lib\site-packages\keras\src\lo
sses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use t
f.compat.v1.losses.sparse_softmax_cross_entropy instead.
```

In [ ]:

```
# you need the current working directory NB: works both windows and linux
current_working_directory = os.getcwd()
current_working_directory = os.path.dirname(current_working_directory)

if not os.path.exists(f"{current_working_directory}/Datasets"):
    os.makedirs(f"{current_working_directory}/Datasets")

print(f"[DATASET] PUT THE DATASET here: {current_working_directory}/Datasets")
```

```
[DATASET] PUT THE DATASET here: C:\Users\Faiza Anan Noor\Computer Vision UTU/Datasets
```

In [ ]:

```
# get the directory where I want to download the dataset
path_of_dataset = os.path.join(*['..', current_working_directory, 'Datasets', 'pizza_not
_pizza'])
print(f"[DIR] The directory of the current dataset is {path_of_dataset}")
```

```
[DIR] The directory of the current dataset is C:\Users\Faiza Anan Noor\Computer Vision UT
U\Datasets\pizza_not_pizza
```

Data prep

In [ ]:

```
# here let s do some functions that we can re-use also for other assignment
def load_the_data_and_the_labels(data_set_path: str, target_size: tuple or None = None):
    """
    This function help you to load the data dynamically
    :param data_set_path: (str) put the path created in the previous cell (is the dataset
path)
    :param target_size: (tuple) the desired size of the images
    :return:
        - array of images
        - array with labels
        - list of labels name (this is used for better visualization)
    """
    try:
        dataset, labels, name_of_the_labels = list(), list(), list()
```

```
        # let s loop here and we try to discover how many class we have
        for class_number, class_name in enumerate(os.listdir(data_set_path)):
            full_path_the_data = os.path.join(*[data_set_path, class_name])
            print(f"[WALK] I am walking into {full_path_the_data}")

            # add the list to nam _list
            name_of_the_labels.append(class_name)

            for single_image in os.listdir(f"{full_path_the_data}"):
                full_path_to_image = os.path.join(*[full_path_the_data, single_image])

                # add the class number
                labels.append(class_number)

                if target_size is None:
                    # let s load the image
                    image = tf.keras.utils.load_img(full_path_to_image)
                else:
                    image = tf.keras.utils.load_img(full_path_to_image, target_size=targ
et_size)

                # transform PIL object in image
                image = tf.keras.utils.img_to_array(image)

                # add the image to the ds list
                dataset.append(image)
      #  print(dataset)

        return np.array(dataset, dtype='uint8'), np.array(labels, dtype='int'), name_of_
the_labels
    except Exception as ex:
        print(f"[EXCEPTION] load the data and the labels throws exceptions {ex}")
```

Load the data

In [ ]:

```
# load the data

dataset, labels, label_names=load_the_data_and_the_labels(path_of_dataset, (224,224))
```

```
[WALK] I am walking into C:\Users\Faiza Anan Noor\Computer Vision UTU\Datasets\pizza_not_
pizza\not_pizza
[WALK] I am walking into C:\Users\Faiza Anan Noor\Computer Vision UTU\Datasets\pizza_not_
pizza\pizza
```

In [ ]:

```
print(labels)
print(label_names)
```

```
[0 0 0 ... 1 1 1]
['not_pizza', 'pizza']
```

In [ ]:

```
label_names[0]
```

Out[ ]:

```
'not_pizza'
```

Normalize the data

In this step, we normalize the dataset by dividing the matrix that containts the image pixel values by 255.0

In [ ]:

```
# normalize the data
```

```
dataset=dataset/255.0
```

Split the data use the train_test_split function

We divide the data into train and test labels and targets using the train_test_split function defining the test size as 30% of the whole data.

In [ ]:

```
# split the data in train and test sets
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(dataset, labels, test_size=0.3, rand
om_state=42)

# Print the shape of the training and testing sets for verification
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (1376, 224, 224, 3)
X_test shape: (590, 224, 224, 3)
y_train shape: (1376,)
y_test shape: (590,)
```

In [ ]:

```
#X_train, X_test, y_train, y_test = X_train[:500], X_test[:200], y_train[:500], y_test[:2
00]
#X_train, X_test, y_train, y_test = X_train[:50], X_test[:20], y_train[:50], y_test[:20]
#X_train, X_test, y_train, y_test = X_train, X_test, y_train, y_test
```

Create the CNN according the instruction:

```
    a. Input layer
    b. Data augmentation, with random flip (horizontal and vertical) and random ro
tation (0.2).
    c. Two hidden layers each composed with the following characteristics: 16 conv
2d units, max pooling 2d and batch normalization, the second one should have 24 con
v2d units max pooling 2d and batch normalization.
    d. After this, add a flatten layer and a dense layer with 8 units
    e. Add the final classifier (a  dense layer) with the correct number of output
and activation
```

In the following code snippet, we create the CNN model according to the instructions above. We defined the input shape as (224 ,224, 3) as target size of the images use (224, 224) and has 3 color channels. Then we defined Sequential model in which we defined layers and actions according to the instructions. To do Data Augmentation(Synthetic Data creation for robust model building, we used the ImageDataGenerator Class in which we passed in necessary random flips(horizontal and vertical both) and defined a random rotation as 0.2 and passed in these parameters into the ImageDataGenerator object called datagen.

Afterwards, we defined two hidden layers. For the first one, we defined 16 conv2d units, max pooling 2d and batch normalization, and also the second one has 24 conv2d units max pooling 2d and batch normalization. After this, we added a flatten layer and a dense layer with 8 units. Finally, we add the final classifier (a dense layer) with the correct number of output and activation, which in our case, the correct number of output is 1 and activation function is sigmoid because we will be working with binary classification(pizza and not pizza).

In [ ]:

```
# Creation of the cnn
# Definition of the input shape (3 channels for color images)
```

```
input_shape = (224, 224, 3)

# Initialization of the model
model = Sequential()

# a) Input layer
model.add(Input(shape=input_shape))

# b) Data augmentation, with random flip and random rotation.

datagen = ImageDataGenerator(
   # rescale=1./255,
    horizontal_flip=True,
    vertical_flip=True,
    rotation_range=0.2
)

# c) Two hidden layers
# 1st hidden layer
model.add(Conv2D(16, (3, 3), activation='relu', input_shape=input_shape))
# Implementation of MaxPool2d
model.add(MaxPooling2D((2, 2)))
model.add(BatchNormalization())

# 2nd hidden layer
model.add(Conv2D(24, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(BatchNormalization())


# d Flatten layer and a dense layer
model.add(Flatten())
model.add(Dense(8, activation='relu'))

# e) Final classifier (dense layer)
# We Replaced 'num_classes' with the correct number of output classes which is 1 in our c
ase
num_classes = 1

# Sigmoid is typically used for binary classification problems.
# For multi, softmax is used. Since ours is binary we used sigmoid
# # Only 1 output neuron. It will contain a value from 0-1 where 1 for 1 class ('pizza')
and 0 for the other ('not pizza')
model.add(Dense(num_classes, activation='sigmoid'))
```

```
WARNING:tensorflow:From C:\Users\Faiza Anan Noor\anaconda3\Lib\site-packages\keras\src\ba
ckend.py:873: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_de
fault_graph instead.

WARNING:tensorflow:From C:\Users\Faiza Anan Noor\anaconda3\Lib\site-packages\keras\src\la
yers\pooling\max_pooling2d.py:161: The name tf.nn.max_pool is deprecated. Please use tf.n
n.max_pool2d instead.
```

compile the model

We Compile the model with Adam optimizer and binary cross entropy as loss function. We also defined the metric to define the model effectiveness as accuracy, and when defining the optimizer, we gave a learning rate of 3e-5. And also we printed out the model summary for our first model.

In [ ]:

```
# compile the CNN
model.compile(
    loss = tf.keras.losses.BinaryCrossentropy(from_logits = True),
    optimizer = tf.keras.optimizers.Adam(learning_rate = 3e-5),
    metrics =["accuracy"],
    )
# Print the model summary
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 222, 222, 16)      448       
                                                                 
 max_pooling2d (MaxPooling2  (None, 111, 111, 16)      0         
 D)                                                              
                                                                 
 batch_normalization (Batch  (None, 111, 111, 16)      64        
 Normalization)                                                  
                                                                 
 conv2d_1 (Conv2D)           (None, 109, 109, 24)      3480      
                                                                 
 max_pooling2d_1 (MaxPoolin  (None, 54, 54, 24)        0         
 g2D)                                                            
                                                                 
 batch_normalization_1 (Bat  (None, 54, 54, 24)        96        
 chNormalization)                                                
                                                                 
 flatten (Flatten)           (None, 69984)             0         
                                                                 
 dense (Dense)               (None, 8)                 559880    
                                                                 
 dense_1 (Dense)             (None, 1)                 9         
                                                                 
=================================================================
Total params: 563977 (2.15 MB)
Trainable params: 563897 (2.15 MB)
Non-trainable params: 80 (320.00 Byte)
_________________________________________________________________
```

We also used Callback to periodically save the Keras model or model weights.

The ModelCheckpoint callback is utilized in combination with model.fit() training to save a model or weights at regular intervals (in a checkpoint file). This allows the model or weights to be loaded at a later time to resume training from the saved state.

In [ ]:

```
checkpoint_path = "training_first_model/cp.ckpt"
checkpoint_dir = os.path.dirname(checkpoint_path)

# Create a callback that saves the model's weights
cp_callback = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_path,
                                                 save_weights_only=True,
                                                 verbose=1)
```

Then using fit() function, we trained the model for 28 epochs, batch size 64 and passed in our callbacks, validation data and defined where the data will flow for training using X_train, y_train(targets and labels).

In [ ]:

```
### Train the model with 128 epochs and 64 batch size batch_size=64
epochs=128
batch_size=64

# Train the model
history = model.fit(
    datagen.flow(X_train, y_train, batch_size=batch_size),
    epochs=epochs,
    validation_data=(X_test, y_test),
    callbacks=[cp_callback],
    verbose=1
)
```

```
Epoch 1/128
```

```
C:\Users\Faiza Anan Noor\anaconda3\Lib\site-packages\keras\src\backend.py:5818: UserWarni
```

```
ng: "binary_crossentropy" received `from_logits=True`, but the `output` argument was pro
duced by a Sigmoid activation and thus does not represent logits. Was this intended?
  output, from_logits = _get_logits(
```

```
WARNING:tensorflow:From C:\Users\Faiza Anan Noor\anaconda3\Lib\site-packages\keras\src\ut
ils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.co
mpat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\Faiza Anan Noor\anaconda3\Lib\site-packages\keras\src\en
gine\base_layer_utils.py:384: The name tf.executing_eagerly_outside_functions is deprecat
ed. Please use tf.compat.v1.executing_eagerly_outside_functions instead.

22/22 [==============================] - ETA: 0s - loss: 0.7493 - accuracy: 0.6265
Epoch 1: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 38s 2s/step - loss: 0.7493 - accuracy: 0.6265 -
val_loss: 0.6831 - val_accuracy: 0.5254
Epoch 2/128
22/22 [==============================] - ETA: 0s - loss: 0.6167 - accuracy: 0.6730
Epoch 2: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.6167 - accuracy: 0.6730 -
val_loss: 0.6778 - val_accuracy: 0.6441
Epoch 3/128
22/22 [==============================] - ETA: 0s - loss: 0.5713 - accuracy: 0.7064
Epoch 3: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.5713 - accuracy: 0.7064 -
val_loss: 0.6788 - val_accuracy: 0.6136
Epoch 4/128
22/22 [==============================] - ETA: 0s - loss: 0.5603 - accuracy: 0.7144
Epoch 4: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.5603 - accuracy: 0.7144 -
val_loss: 0.6737 - val_accuracy: 0.6085
Epoch 5/128
22/22 [==============================] - ETA: 0s - loss: 0.5225 - accuracy: 0.7594
Epoch 5: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.5225 - accuracy: 0.7594 -
val_loss: 0.6833 - val_accuracy: 0.5102
Epoch 6/128
22/22 [==============================] - ETA: 0s - loss: 0.4875 - accuracy: 0.7762
Epoch 6: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.4875 - accuracy: 0.7762 -
val_loss: 0.6619 - val_accuracy: 0.6678
Epoch 7/128
22/22 [==============================] - ETA: 0s - loss: 0.4757 - accuracy: 0.7878
Epoch 7: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.4757 - accuracy: 0.7878 -
val_loss: 0.6536 - val_accuracy: 0.6492
Epoch 8/128
22/22 [==============================] - ETA: 0s - loss: 0.4524 - accuracy: 0.7892
Epoch 8: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.4524 - accuracy: 0.7892 -
val_loss: 0.6438 - val_accuracy: 0.6458
Epoch 9/128
22/22 [==============================] - ETA: 0s - loss: 0.4125 - accuracy: 0.8263
Epoch 9: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.4125 - accuracy: 0.8263 -
val_loss: 0.6515 - val_accuracy: 0.6169
Epoch 10/128
22/22 [==============================] - ETA: 0s - loss: 0.4026 - accuracy: 0.8307
Epoch 10: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.4026 - accuracy: 0.8307 -
val_loss: 0.6469 - val_accuracy: 0.5864
Epoch 11/128
22/22 [==============================] - ETA: 0s - loss: 0.3716 - accuracy: 0.8343
Epoch 11: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.3716 - accuracy: 0.8343 -
val_loss: 0.6264 - val_accuracy: 0.6305
Epoch 12/128
22/22 [==============================] - ETA: 0s - loss: 0.3631 - accuracy: 0.8321
Epoch 12: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.3631 - accuracy: 0.8321 -
val_loss: 0.6959 - val_accuracy: 0.5424
Epoch 13/128
22/22 [==============================] - ETA: 0s - loss: 0.3499 - accuracy: 0.8590
```

```
Epoch 13: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.3499 - accuracy: 0.8590 -
val_loss: 0.6310 - val_accuracy: 0.6237
Epoch 14/128
22/22 [==============================] - ETA: 0s - loss: 0.3326 - accuracy: 0.8677
Epoch 14: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.3326 - accuracy: 0.8677 -
val_loss: 0.6490 - val_accuracy: 0.6000
Epoch 15/128
22/22 [==============================] - ETA: 0s - loss: 0.3218 - accuracy: 0.8685
Epoch 15: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.3218 - accuracy: 0.8685 -
val_loss: 0.6304 - val_accuracy: 0.6237
Epoch 16/128
22/22 [==============================] - ETA: 0s - loss: 0.3120 - accuracy: 0.8823
Epoch 16: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.3120 - accuracy: 0.8823 -
val_loss: 0.6613 - val_accuracy: 0.6119
Epoch 17/128
22/22 [==============================] - ETA: 0s - loss: 0.3040 - accuracy: 0.8757
Epoch 17: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.3040 - accuracy: 0.8757 -
val_loss: 0.6348 - val_accuracy: 0.6356
Epoch 18/128
22/22 [==============================] - ETA: 0s - loss: 0.2721 - accuracy: 0.8924
Epoch 18: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.2721 - accuracy: 0.8924 -
val_loss: 0.6917 - val_accuracy: 0.6085
Epoch 19/128
22/22 [==============================] - ETA: 0s - loss: 0.2726 - accuracy: 0.8888
Epoch 19: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.2726 - accuracy: 0.8888 -
val_loss: 0.6686 - val_accuracy: 0.6271
Epoch 20/128
22/22 [==============================] - ETA: 0s - loss: 0.2600 - accuracy: 0.9077
Epoch 20: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.2600 - accuracy: 0.9077 -
val_loss: 0.6621 - val_accuracy: 0.6458
Epoch 21/128
22/22 [==============================] - ETA: 0s - loss: 0.2583 - accuracy: 0.9019
Epoch 21: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.2583 - accuracy: 0.9019 -
val_loss: 0.6426 - val_accuracy: 0.6661
Epoch 22/128
22/22 [==============================] - ETA: 0s - loss: 0.2312 - accuracy: 0.9266
Epoch 22: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.2312 - accuracy: 0.9266 -
val_loss: 0.6190 - val_accuracy: 0.6780
Epoch 23/128
22/22 [==============================] - ETA: 0s - loss: 0.2297 - accuracy: 0.9172
Epoch 23: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 27s 1s/step - loss: 0.2297 - accuracy: 0.9172 -
val_loss: 0.6232 - val_accuracy: 0.6881
Epoch 24/128
22/22 [==============================] - ETA: 0s - loss: 0.2140 - accuracy: 0.9331
Epoch 24: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 27s 1s/step - loss: 0.2140 - accuracy: 0.9331 -
val_loss: 0.5858 - val_accuracy: 0.7288
Epoch 25/128
22/22 [==============================] - ETA: 0s - loss: 0.2143 - accuracy: 0.9259
Epoch 25: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.2143 - accuracy: 0.9259 -
val_loss: 0.6069 - val_accuracy: 0.7169
Epoch 26/128
22/22 [==============================] - ETA: 0s - loss: 0.2073 - accuracy: 0.9331
Epoch 26: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.2073 - accuracy: 0.9331 -
val_loss: 0.6283 - val_accuracy: 0.7136
Epoch 27/128
22/22 [==============================] - ETA: 0s - loss: 0.2065 - accuracy: 0.9288
Epoch 27: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.2065 - accuracy: 0.9288 -
```

```
val_loss: 0.5896 - val_accuracy: 0.7390
Epoch 28/128
22/22 [==============================] - ETA: 0s - loss: 0.1888 - accuracy: 0.9426
Epoch 28: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 27s 1s/step - loss: 0.1888 - accuracy: 0.9426 -
val_loss: 0.6118 - val_accuracy: 0.7271
Epoch 29/128
22/22 [==============================] - ETA: 0s - loss: 0.1823 - accuracy: 0.9440
Epoch 29: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.1823 - accuracy: 0.9440 -
val_loss: 0.6242 - val_accuracy: 0.7254
Epoch 30/128
22/22 [==============================] - ETA: 0s - loss: 0.1765 - accuracy: 0.9491
Epoch 30: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.1765 - accuracy: 0.9491 -
val_loss: 0.6345 - val_accuracy: 0.7186
Epoch 31/128
22/22 [==============================] - ETA: 0s - loss: 0.1703 - accuracy: 0.9513
Epoch 31: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.1703 - accuracy: 0.9513 -
val_loss: 0.6176 - val_accuracy: 0.7288
Epoch 32/128
22/22 [==============================] - ETA: 0s - loss: 0.1778 - accuracy: 0.9448
Epoch 32: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.1778 - accuracy: 0.9448 -
val_loss: 0.6372 - val_accuracy: 0.7305
Epoch 33/128
22/22 [==============================] - ETA: 0s - loss: 0.1622 - accuracy: 0.9542
Epoch 33: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.1622 - accuracy: 0.9542 -
val_loss: 0.6512 - val_accuracy: 0.7322
Epoch 34/128
22/22 [==============================] - ETA: 0s - loss: 0.1543 - accuracy: 0.9535
Epoch 34: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.1543 - accuracy: 0.9535 -
val_loss: 0.6515 - val_accuracy: 0.7271
Epoch 35/128
22/22 [==============================] - ETA: 0s - loss: 0.1461 - accuracy: 0.9651
Epoch 35: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.1461 - accuracy: 0.9651 -
val_loss: 0.6454 - val_accuracy: 0.7322
Epoch 36/128
22/22 [==============================] - ETA: 0s - loss: 0.1461 - accuracy: 0.9622
Epoch 36: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 27s 1s/step - loss: 0.1461 - accuracy: 0.9622 -
val_loss: 0.6810 - val_accuracy: 0.7068
Epoch 37/128
22/22 [==============================] - ETA: 0s - loss: 0.1405 - accuracy: 0.9622
Epoch 37: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.1405 - accuracy: 0.9622 -
val_loss: 0.6732 - val_accuracy: 0.7203
Epoch 38/128
22/22 [==============================] - ETA: 0s - loss: 0.1332 - accuracy: 0.9717
Epoch 38: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.1332 - accuracy: 0.9717 -
val_loss: 0.6841 - val_accuracy: 0.7339
Epoch 39/128
22/22 [==============================] - ETA: 0s - loss: 0.1279 - accuracy: 0.9680
Epoch 39: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.1279 - accuracy: 0.9680 -
val_loss: 0.6856 - val_accuracy: 0.7441
Epoch 40/128
22/22 [==============================] - ETA: 0s - loss: 0.1203 - accuracy: 0.9760
Epoch 40: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 27s 1s/step - loss: 0.1203 - accuracy: 0.9760 -
val_loss: 0.6753 - val_accuracy: 0.7339
Epoch 41/128
22/22 [==============================] - ETA: 0s - loss: 0.1206 - accuracy: 0.9746
Epoch 41: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.1206 - accuracy: 0.9746 -
val_loss: 0.6842 - val_accuracy: 0.7373
Epoch 42/128
```

```
22/22 [==============================] - ETA: 0s - loss: 0.1106 - accuracy: 0.9775
Epoch 42: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1106 - accuracy: 0.9775 -
val_loss: 0.6797 - val_accuracy: 0.7441
Epoch 43/128
22/22 [==============================] - ETA: 0s - loss: 0.1065 - accuracy: 0.9767
Epoch 43: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1065 - accuracy: 0.9767 -
val_loss: 0.6923 - val_accuracy: 0.7356
Epoch 44/128
22/22 [==============================] - ETA: 0s - loss: 0.1051 - accuracy: 0.9804
Epoch 44: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1051 - accuracy: 0.9804 -
val_loss: 0.7000 - val_accuracy: 0.7339
Epoch 45/128
22/22 [==============================] - ETA: 0s - loss: 0.1034 - accuracy: 0.9826
Epoch 45: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.1034 - accuracy: 0.9826 -
val_loss: 0.7222 - val_accuracy: 0.7254
Epoch 46/128
22/22 [==============================] - ETA: 0s - loss: 0.1008 - accuracy: 0.9811
Epoch 46: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1008 - accuracy: 0.9811 -
val_loss: 0.7228 - val_accuracy: 0.7288
Epoch 47/128
22/22 [==============================] - ETA: 0s - loss: 0.0932 - accuracy: 0.9847
Epoch 47: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0932 - accuracy: 0.9847 -
val_loss: 0.7019 - val_accuracy: 0.7305
Epoch 48/128
22/22 [==============================] - ETA: 0s - loss: 0.0928 - accuracy: 0.9847
Epoch 48: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0928 - accuracy: 0.9847 -
val_loss: 0.7148 - val_accuracy: 0.7390
Epoch 49/128
22/22 [==============================] - ETA: 0s - loss: 0.0900 - accuracy: 0.9826
Epoch 49: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0900 - accuracy: 0.9826 -
val_loss: 0.7259 - val_accuracy: 0.7305
Epoch 50/128
22/22 [==============================] - ETA: 0s - loss: 0.0922 - accuracy: 0.9826
Epoch 50: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0922 - accuracy: 0.9826 -
val_loss: 0.7229 - val_accuracy: 0.7339
Epoch 51/128
22/22 [==============================] - ETA: 0s - loss: 0.0831 - accuracy: 0.9862
Epoch 51: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0831 - accuracy: 0.9862 -
val_loss: 0.7311 - val_accuracy: 0.7373
Epoch 52/128
22/22 [==============================] - ETA: 0s - loss: 0.0780 - accuracy: 0.9862
Epoch 52: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0780 - accuracy: 0.9862 -
val_loss: 0.7282 - val_accuracy: 0.7356
Epoch 53/128
22/22 [==============================] - ETA: 0s - loss: 0.0790 - accuracy: 0.9898
Epoch 53: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0790 - accuracy: 0.9898 -
val_loss: 0.7416 - val_accuracy: 0.7424
Epoch 54/128
22/22 [==============================] - ETA: 0s - loss: 0.0737 - accuracy: 0.9906
Epoch 54: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0737 - accuracy: 0.9906 -
val_loss: 0.7351 - val_accuracy: 0.7339
Epoch 55/128
22/22 [==============================] - ETA: 0s - loss: 0.0745 - accuracy: 0.9862
Epoch 55: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0745 - accuracy: 0.9862 -
val_loss: 0.7220 - val_accuracy: 0.7339
Epoch 56/128
22/22 [==============================] - ETA: 0s - loss: 0.0684 - accuracy: 0.9920
Epoch 56: saving model to training_first_model\cp.ckpt
```

```
22/22 [==============================] - 30s 1s/step - loss: 0.0684 - accuracy: 0.9920 -
val_loss: 0.7428 - val_accuracy: 0.7305
Epoch 57/128
22/22 [==============================] - ETA: 0s - loss: 0.0693 - accuracy: 0.9913
Epoch 57: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0693 - accuracy: 0.9913 -
val_loss: 0.7228 - val_accuracy: 0.7373
Epoch 58/128
22/22 [==============================] - ETA: 0s - loss: 0.0645 - accuracy: 0.9913
Epoch 58: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0645 - accuracy: 0.9913 -
val_loss: 0.7341 - val_accuracy: 0.7339
Epoch 59/128
22/22 [==============================] - ETA: 0s - loss: 0.0668 - accuracy: 0.9906
Epoch 59: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0668 - accuracy: 0.9906 -
val_loss: 0.7294 - val_accuracy: 0.7339
Epoch 60/128
22/22 [==============================] - ETA: 0s - loss: 0.0640 - accuracy: 0.9906
Epoch 60: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0640 - accuracy: 0.9906 -
val_loss: 0.7456 - val_accuracy: 0.7254
Epoch 61/128
22/22 [==============================] - ETA: 0s - loss: 0.0645 - accuracy: 0.9906
Epoch 61: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0645 - accuracy: 0.9906 -
val_loss: 0.7410 - val_accuracy: 0.7441
Epoch 62/128
22/22 [==============================] - ETA: 0s - loss: 0.0613 - accuracy: 0.9913
Epoch 62: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0613 - accuracy: 0.9913 -
val_loss: 0.7588 - val_accuracy: 0.7322
Epoch 63/128
22/22 [==============================] - ETA: 0s - loss: 0.0560 - accuracy: 0.9927
Epoch 63: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0560 - accuracy: 0.9927 -
val_loss: 0.7560 - val_accuracy: 0.7254
Epoch 64/128
22/22 [==============================] - ETA: 0s - loss: 0.0603 - accuracy: 0.9920
Epoch 64: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0603 - accuracy: 0.9920 -
val_loss: 0.7563 - val_accuracy: 0.7271
Epoch 65/128
22/22 [==============================] - ETA: 0s - loss: 0.0522 - accuracy: 0.9956
Epoch 65: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0522 - accuracy: 0.9956 -
val_loss: 0.7383 - val_accuracy: 0.7407
Epoch 66/128
22/22 [==============================] - ETA: 0s - loss: 0.0525 - accuracy: 0.9949
Epoch 66: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.0525 - accuracy: 0.9949 -
val_loss: 0.7481 - val_accuracy: 0.7356
Epoch 67/128
22/22 [==============================] - ETA: 0s - loss: 0.0498 - accuracy: 0.9949
Epoch 67: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0498 - accuracy: 0.9949 -
val_loss: 0.7441 - val_accuracy: 0.7356
Epoch 68/128
22/22 [==============================] - ETA: 0s - loss: 0.0527 - accuracy: 0.9949
Epoch 68: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0527 - accuracy: 0.9949 -
val_loss: 0.7793 - val_accuracy: 0.7322
Epoch 69/128
22/22 [==============================] - ETA: 0s - loss: 0.0501 - accuracy: 0.9935
Epoch 69: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0501 - accuracy: 0.9935 -
val_loss: 0.7548 - val_accuracy: 0.7288
Epoch 70/128
22/22 [==============================] - ETA: 0s - loss: 0.0483 - accuracy: 0.9949
Epoch 70: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0483 - accuracy: 0.9949 -
val_loss: 0.7498 - val_accuracy: 0.7288
```

```
Epoch 71/128
22/22 [==============================] - ETA: 0s - loss: 0.0470 - accuracy: 0.9949
Epoch 71: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0470 - accuracy: 0.9949 -
val_loss: 0.7576 - val_accuracy: 0.7305
Epoch 72/128
22/22 [==============================] - ETA: 0s - loss: 0.0468 - accuracy: 0.9956
Epoch 72: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0468 - accuracy: 0.9956 -
val_loss: 0.7556 - val_accuracy: 0.7356
Epoch 73/128
22/22 [==============================] - ETA: 0s - loss: 0.0444 - accuracy: 0.9971
Epoch 73: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0444 - accuracy: 0.9971 -
val_loss: 0.7536 - val_accuracy: 0.7407
Epoch 74/128
22/22 [==============================] - ETA: 0s - loss: 0.0423 - accuracy: 0.9956
Epoch 74: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0423 - accuracy: 0.9956 -
val_loss: 0.7615 - val_accuracy: 0.7356
Epoch 75/128
22/22 [==============================] - ETA: 0s - loss: 0.0415 - accuracy: 0.9956
Epoch 75: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0415 - accuracy: 0.9956 -
val_loss: 0.7636 - val_accuracy: 0.7424
Epoch 76/128
22/22 [==============================] - ETA: 0s - loss: 0.0399 - accuracy: 0.9985
Epoch 76: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0399 - accuracy: 0.9985 -
val_loss: 0.7522 - val_accuracy: 0.7458
Epoch 77/128
22/22 [==============================] - ETA: 0s - loss: 0.0380 - accuracy: 0.9978
Epoch 77: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0380 - accuracy: 0.9978 -
val_loss: 0.7642 - val_accuracy: 0.7407
Epoch 78/128
22/22 [==============================] - ETA: 0s - loss: 0.0389 - accuracy: 0.9949
Epoch 78: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 27s 1s/step - loss: 0.0389 - accuracy: 0.9949 -
val_loss: 0.7702 - val_accuracy: 0.7322
Epoch 79/128
22/22 [==============================] - ETA: 0s - loss: 0.0360 - accuracy: 0.9985
Epoch 79: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 27s 1s/step - loss: 0.0360 - accuracy: 0.9985 -
val_loss: 0.7718 - val_accuracy: 0.7424
Epoch 80/128
22/22 [==============================] - ETA: 0s - loss: 0.0386 - accuracy: 0.9978
Epoch 80: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 27s 1s/step - loss: 0.0386 - accuracy: 0.9978 -
val_loss: 0.7891 - val_accuracy: 0.7390
Epoch 81/128
22/22 [==============================] - ETA: 0s - loss: 0.0391 - accuracy: 0.9956
Epoch 81: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 26s 1s/step - loss: 0.0391 - accuracy: 0.9956 -
val_loss: 0.7677 - val_accuracy: 0.7288
Epoch 82/128
22/22 [==============================] - ETA: 0s - loss: 0.0362 - accuracy: 0.9971
Epoch 82: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.0362 - accuracy: 0.9971 -
val_loss: 0.7762 - val_accuracy: 0.7339
Epoch 83/128
22/22 [==============================] - ETA: 0s - loss: 0.0354 - accuracy: 0.9978
Epoch 83: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.0354 - accuracy: 0.9978 -
val_loss: 0.7679 - val_accuracy: 0.7390
Epoch 84/128
22/22 [==============================] - ETA: 0s - loss: 0.0354 - accuracy: 0.9978
Epoch 84: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0354 - accuracy: 0.9978 -
val_loss: 0.7833 - val_accuracy: 0.7424
Epoch 85/128
22/22 [==============================] - ETA: 0s - loss: 0.0349 - accuracy: 0.9964
```

```
Epoch 85: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.0349 - accuracy: 0.9964 -
val_loss: 0.7786 - val_accuracy: 0.7254
Epoch 86/128
22/22 [==============================] - ETA: 0s - loss: 0.0315 - accuracy: 0.9964
Epoch 86: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0315 - accuracy: 0.9964 -
val_loss: 0.7838 - val_accuracy: 0.7271
Epoch 87/128
22/22 [==============================] - ETA: 0s - loss: 0.0314 - accuracy: 0.9985
Epoch 87: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.0314 - accuracy: 0.9985 -
val_loss: 0.7847 - val_accuracy: 0.7390
Epoch 88/128
22/22 [==============================] - ETA: 0s - loss: 0.0307 - accuracy: 0.9978
Epoch 88: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.0307 - accuracy: 0.9978 -
val_loss: 0.7871 - val_accuracy: 0.7271
Epoch 89/128
22/22 [==============================] - ETA: 0s - loss: 0.0296 - accuracy: 0.9978
Epoch 89: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0296 - accuracy: 0.9978 -
val_loss: 0.7886 - val_accuracy: 0.7373
Epoch 90/128
22/22 [==============================] - ETA: 0s - loss: 0.0295 - accuracy: 0.9978
Epoch 90: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0295 - accuracy: 0.9978 -
val_loss: 0.7901 - val_accuracy: 0.7237
Epoch 91/128
22/22 [==============================] - ETA: 0s - loss: 0.0313 - accuracy: 0.9971
Epoch 91: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0313 - accuracy: 0.9971 -
val_loss: 0.7925 - val_accuracy: 0.7390
Epoch 92/128
22/22 [==============================] - ETA: 0s - loss: 0.0272 - accuracy: 0.9985
Epoch 92: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0272 - accuracy: 0.9985 -
val_loss: 0.8077 - val_accuracy: 0.7407
Epoch 93/128
22/22 [==============================] - ETA: 0s - loss: 0.0285 - accuracy: 0.9993
Epoch 93: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0285 - accuracy: 0.9993 -
val_loss: 0.8149 - val_accuracy: 0.7424
Epoch 94/128
22/22 [==============================] - ETA: 0s - loss: 0.0273 - accuracy: 0.9971
Epoch 94: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0273 - accuracy: 0.9971 -
val_loss: 0.8042 - val_accuracy: 0.7322
Epoch 95/128
22/22 [==============================] - ETA: 0s - loss: 0.0277 - accuracy: 0.9985
Epoch 95: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0277 - accuracy: 0.9985 -
val_loss: 0.7955 - val_accuracy: 0.7458
Epoch 96/128
22/22 [==============================] - ETA: 0s - loss: 0.0248 - accuracy: 0.9993
Epoch 96: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0248 - accuracy: 0.9993 -
val_loss: 0.8069 - val_accuracy: 0.7373
Epoch 97/128
22/22 [==============================] - ETA: 0s - loss: 0.0261 - accuracy: 0.9971
Epoch 97: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0261 - accuracy: 0.9971 -
val_loss: 0.8075 - val_accuracy: 0.7339
Epoch 98/128
22/22 [==============================] - ETA: 0s - loss: 0.0251 - accuracy: 0.9985
Epoch 98: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0251 - accuracy: 0.9985 -
val_loss: 0.8169 - val_accuracy: 0.7288
Epoch 99/128
22/22 [==============================] - ETA: 0s - loss: 0.0222 - accuracy: 0.9993
Epoch 99: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.0222 - accuracy: 0.9993 -
```

```
val_loss: 0.8104 - val_accuracy: 0.7254
Epoch 100/128
22/22 [==============================] - ETA: 0s - loss: 0.0221 - accuracy: 0.9993
Epoch 100: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.0221 - accuracy: 0.9993 -
val_loss: 0.8206 - val_accuracy: 0.7322
Epoch 101/128
22/22 [==============================] - ETA: 0s - loss: 0.0244 - accuracy: 0.9985
Epoch 101: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0244 - accuracy: 0.9985 -
val_loss: 0.8129 - val_accuracy: 0.7288
Epoch 102/128
22/22 [==============================] - ETA: 0s - loss: 0.0226 - accuracy: 0.9978
Epoch 102: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0226 - accuracy: 0.9978 -
val_loss: 0.8050 - val_accuracy: 0.7322
Epoch 103/128
22/22 [==============================] - ETA: 0s - loss: 0.0217 - accuracy: 1.0000
Epoch 103: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0217 - accuracy: 1.0000 -
val_loss: 0.8012 - val_accuracy: 0.7441
Epoch 104/128
22/22 [==============================] - ETA: 0s - loss: 0.0228 - accuracy: 0.9985
Epoch 104: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0228 - accuracy: 0.9985 -
val_loss: 0.8139 - val_accuracy: 0.7356
Epoch 105/128
22/22 [==============================] - ETA: 0s - loss: 0.0208 - accuracy: 0.9993
Epoch 105: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0208 - accuracy: 0.9993 -
val_loss: 0.8097 - val_accuracy: 0.7373
Epoch 106/128
22/22 [==============================] - ETA: 0s - loss: 0.0218 - accuracy: 0.9971
Epoch 106: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0218 - accuracy: 0.9971 -
val_loss: 0.8084 - val_accuracy: 0.7373
Epoch 107/128
22/22 [==============================] - ETA: 0s - loss: 0.0208 - accuracy: 0.9993
Epoch 107: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0208 - accuracy: 0.9993 -
val_loss: 0.8141 - val_accuracy: 0.7339
Epoch 108/128
22/22 [==============================] - ETA: 0s - loss: 0.0222 - accuracy: 0.9956
Epoch 108: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0222 - accuracy: 0.9956 -
val_loss: 0.8035 - val_accuracy: 0.7407
Epoch 109/128
22/22 [==============================] - ETA: 0s - loss: 0.0210 - accuracy: 1.0000
Epoch 109: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0210 - accuracy: 1.0000 -
val_loss: 0.8101 - val_accuracy: 0.7356
Epoch 110/128
22/22 [==============================] - ETA: 0s - loss: 0.0215 - accuracy: 0.9978
Epoch 110: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0215 - accuracy: 0.9978 -
val_loss: 0.8165 - val_accuracy: 0.7271
Epoch 111/128
22/22 [==============================] - ETA: 0s - loss: 0.0188 - accuracy: 0.9993
Epoch 111: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.0188 - accuracy: 0.9993 -
val_loss: 0.8145 - val_accuracy: 0.7373
Epoch 112/128
22/22 [==============================] - ETA: 0s - loss: 0.0197 - accuracy: 1.0000
Epoch 112: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0197 - accuracy: 1.0000 -
val_loss: 0.8189 - val_accuracy: 0.7271
Epoch 113/128
22/22 [==============================] - ETA: 0s - loss: 0.0185 - accuracy: 0.9993
Epoch 113: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.0185 - accuracy: 0.9993 -
val_loss: 0.8350 - val_accuracy: 0.7203
Epoch 114/128
```

```
22/22 [==============================] - ETA: 0s - loss: 0.0167 - accuracy: 0.9993
Epoch 114: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0167 - accuracy: 0.9993 -
val_loss: 0.8252 - val_accuracy: 0.7424
Epoch 115/128
22/22 [==============================] - ETA: 0s - loss: 0.0172 - accuracy: 0.9985
Epoch 115: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0172 - accuracy: 0.9985 -
val_loss: 0.8273 - val_accuracy: 0.7356
Epoch 116/128
22/22 [==============================] - ETA: 0s - loss: 0.0163 - accuracy: 1.0000
Epoch 116: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0163 - accuracy: 1.0000 -
val_loss: 0.8234 - val_accuracy: 0.7441
Epoch 117/128
22/22 [==============================] - ETA: 0s - loss: 0.0164 - accuracy: 1.0000
Epoch 117: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0164 - accuracy: 1.0000 -
val_loss: 0.8310 - val_accuracy: 0.7424
Epoch 118/128
22/22 [==============================] - ETA: 0s - loss: 0.0168 - accuracy: 0.9993
Epoch 118: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0168 - accuracy: 0.9993 -
val_loss: 0.8194 - val_accuracy: 0.7373
Epoch 119/128
22/22 [==============================] - ETA: 0s - loss: 0.0159 - accuracy: 0.9993
Epoch 119: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0159 - accuracy: 0.9993 -
val_loss: 0.8316 - val_accuracy: 0.7339
Epoch 120/128
22/22 [==============================] - ETA: 0s - loss: 0.0162 - accuracy: 0.9985
Epoch 120: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.0162 - accuracy: 0.9985 -
val_loss: 0.8335 - val_accuracy: 0.7407
Epoch 121/128
22/22 [==============================] - ETA: 0s - loss: 0.0159 - accuracy: 0.9978
Epoch 121: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0159 - accuracy: 0.9978 -
val_loss: 0.8285 - val_accuracy: 0.7390
Epoch 122/128
22/22 [==============================] - ETA: 0s - loss: 0.0146 - accuracy: 1.0000
Epoch 122: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0146 - accuracy: 1.0000 -
val_loss: 0.8355 - val_accuracy: 0.7441
Epoch 123/128
22/22 [==============================] - ETA: 0s - loss: 0.0148 - accuracy: 0.9993
Epoch 123: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0148 - accuracy: 0.9993 -
val_loss: 0.8352 - val_accuracy: 0.7390
Epoch 124/128
22/22 [==============================] - ETA: 0s - loss: 0.0150 - accuracy: 1.0000
Epoch 124: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0150 - accuracy: 1.0000 -
val_loss: 0.8474 - val_accuracy: 0.7373
Epoch 125/128
22/22 [==============================] - ETA: 0s - loss: 0.0149 - accuracy: 1.0000
Epoch 125: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0149 - accuracy: 1.0000 -
val_loss: 0.8432 - val_accuracy: 0.7373
Epoch 126/128
22/22 [==============================] - ETA: 0s - loss: 0.0152 - accuracy: 0.9985
Epoch 126: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0152 - accuracy: 0.9985 -
val_loss: 0.8283 - val_accuracy: 0.7373
Epoch 127/128
22/22 [==============================] - ETA: 0s - loss: 0.0137 - accuracy: 1.0000
Epoch 127: saving model to training_first_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.0137 - accuracy: 1.0000 -
val_loss: 0.8329 - val_accuracy: 0.7458
Epoch 128/128
22/22 [==============================] - ETA: 0s - loss: 0.0146 - accuracy: 0.9993
Epoch 128: saving model to training_first_model\cp.ckpt
```

```
22/22 [==============================] - 29s 1s/step - loss: 0.0146 - accuracy: 0.9993 - val_loss: 0.8286 - val_accuracy: 0.7424
```

In [ ]:

```
# Save the entire model as a `.keras` zip archive.
model.save('First_model.keras')
```

Evaluate the model and report the accuracy

I evaluated the model using model.evaluate passing in our test labels and targets. And then from that variable containing the evaluation results, we take the index 1 to get the accuracy of our model on the test set.

In [ ]:

```
model= tf.keras.models.load_model('First_model.keras')
```

In [ ]:

```
eval_result = model.evaluate(X_test, y_test, verbose=1)

# Print the accuracy
accuracy = eval_result[1] * 100  # Accuracy is typically the second element in the evalua
tion result
print(f'Test Accuracy: {accuracy:.2f}%')#
```

```
19/19 [==============================] - 7s 57ms/step - loss: 0.8286 - accuracy: 0.7424
Test Accuracy: 74.24%
```

For our first model, the accuracy is 74.24% which means that our model has quite a decent accuracy and for 74% of the cases it can accurately classify between the pizza and not pizza classes.

Make prediction with the test set and use a threshold of 0.5 as boundaries decision between the classes.

Then we make prediction using the test set(X_test) and the variable "predictions" contains the results. Then we set threshold as 0.5 and make binary predictions variable contain predictions as 1 for values that are above 0.5 and 0 otherwise. And this is used using the "predictions" variable defined above containing the real values for the predictions. Since we do not have binary predictions as we see by printing the predictions array, we convert them to binary labels and predictions using the specified threshold of 0.5

In [ ]:

```
# Since we do not have binary predictions, we convert them to binary labels and predictio
ns using the specified threshold of 0.5

threshold=0.5
predictions=model.predict(X_test)
predictions[:9]
```

```
19/19 [==============================] - 6s 43ms/step
```

Out[ ]:

```
array([[9.9806821e-01],
       [5.5093551e-01],
       [1.2737493e-03],
       [9.9979383e-01],
       [9.9999976e-01],
       [9.1787595e-01],
       [2.5328493e-01],
       [8.7279412e-05],
       [9.8753053e-01]], dtype=float32)
```

In [ ]:

```
binary_labels = np.argmax(predictions, axis=1)
```

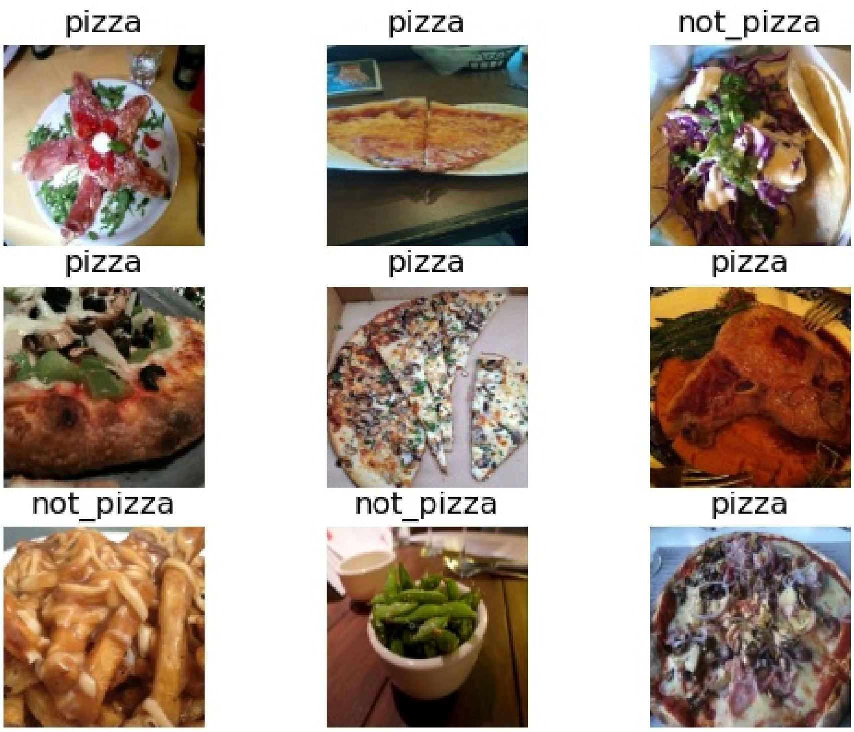
```
binary_predictions = (predictions > threshold).astype(int)
#binary_predictions
```

show predictions

Then we tried to show some predictions by plotting a test image against our predicted Label Results. We can see that almost 74% of the time, it gets correct results. We tried to plot this a couple of times after this code snippet using different subsets of the test set.

In [ ]:

```
# do it here
def show_some_prediction(number_of_subplot, test_set, predictions, name_of_the_labels):
    for i in range(number_of_subplot):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(test_set[i])
        plt.title(f'{name_of_the_labels[predictions[i]]}')
        plt.axis("off")
    plt.show()
show_some_prediction(9, X_test, binary_predictions.flatten(), label_names)
```



In [ ]:

```
show_some_prediction(9, X_test[40:60], binary_predictions.flatten()[40:60], label_names)
```

In [ ]:

```
show_some_prediction(9, X_test[50:60], binary_predictions.flatten()[50:60], label_names)
```

show metrics like confusion matrix or ROC curve or both (sklearn has already implemented all these stuff)

In [ ]:

```
print("Confusion Matrix and ROC curve of model 1 is given below : ")
```

```
Confusion Matrix and ROC curve of model 1 is given below :
```

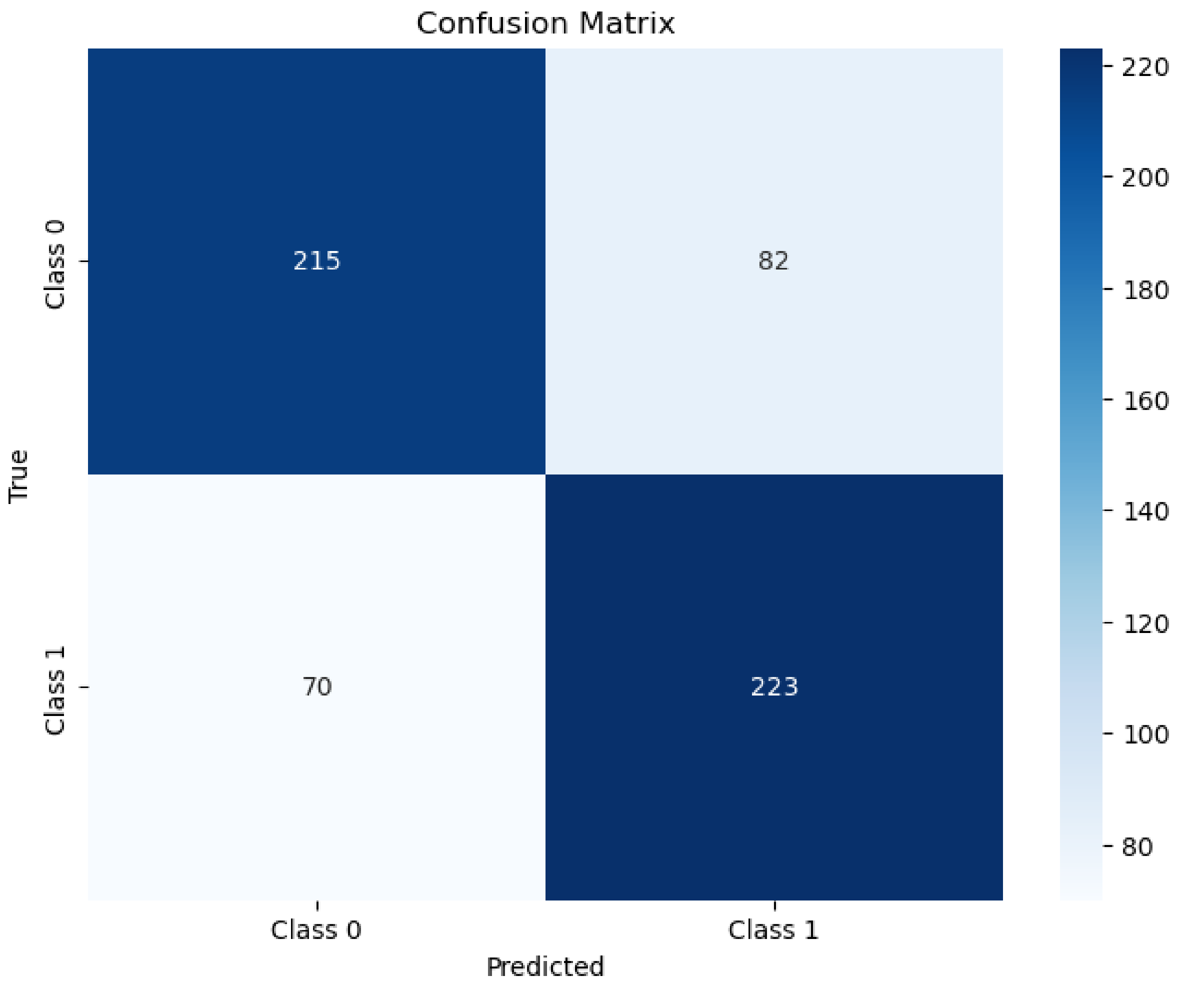
In [ ]:

```

```

In [ ]:

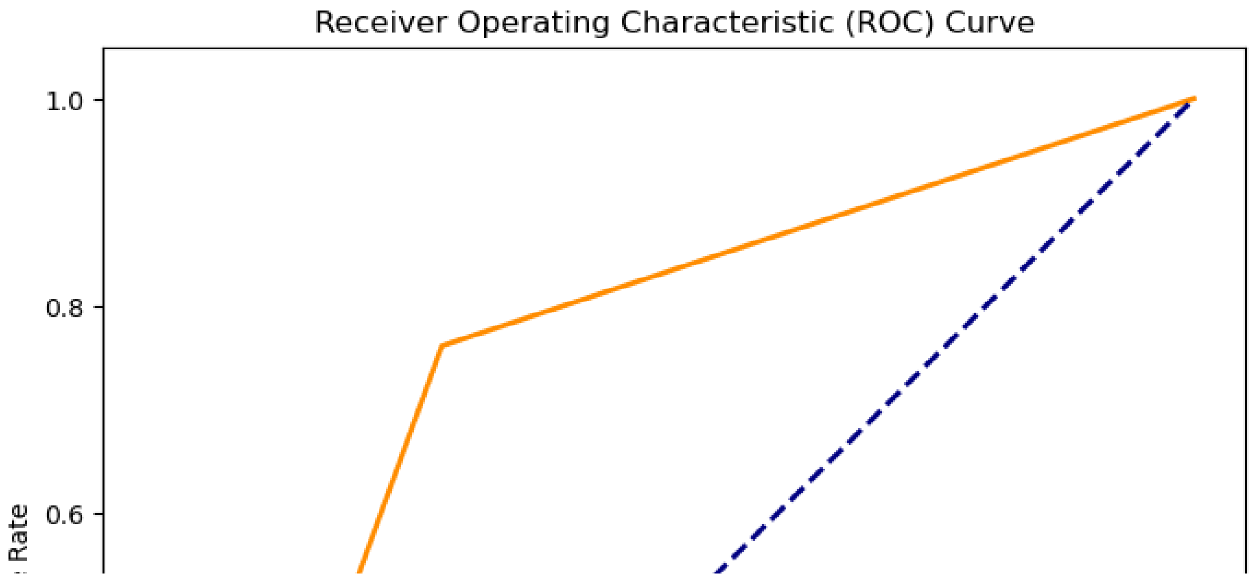
```
# Calculate and display the confusion matrix
conf_matrix = confusion_matrix(y_test,  binary_predictions.flatten())
print("Confusion Matrix:")
print(conf_matrix)

# Plot Confusion Matrix using Matplotlib
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Class 0', 'Cla
ss 1'], yticklabels=['Class 0', 'Class 1'])
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()
```

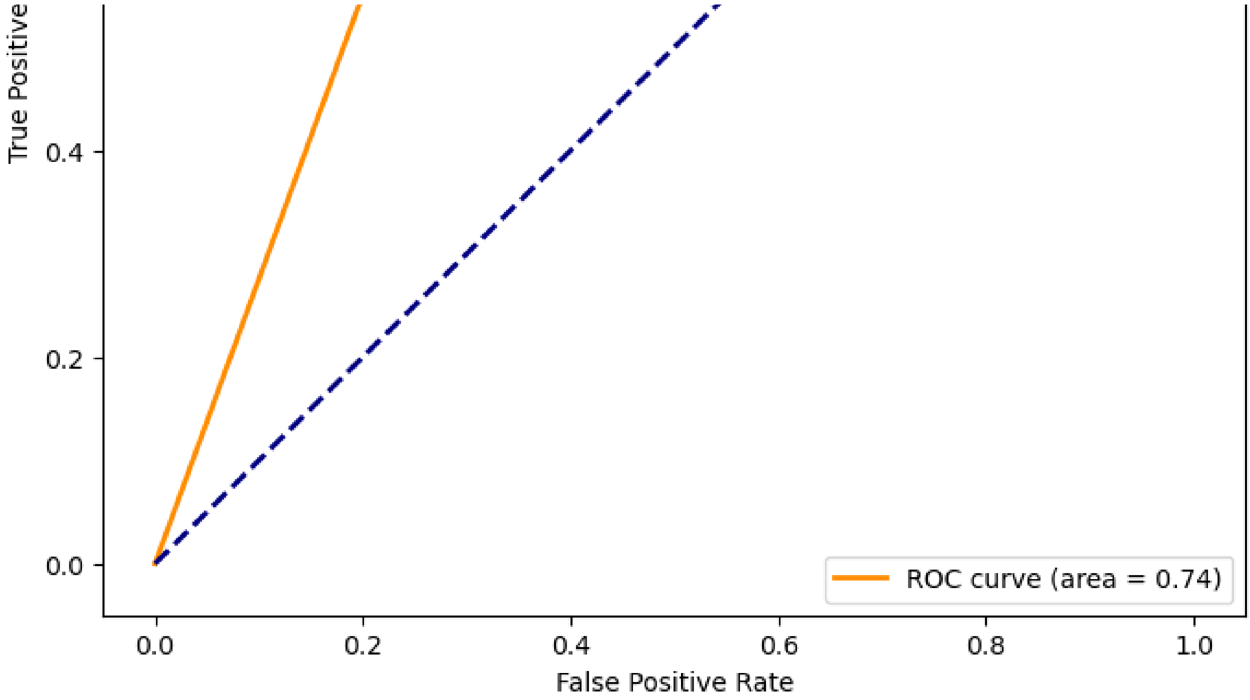
```
Confusion Matrix:
[[215  82]
 [ 70 223]]
```



In [ ]:

```
# Plot ROC Curve
fpr, tpr, thresholds = roc_curve(y_test,  binary_predictions.flatten())
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 8))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()
```





For our first model, the area under the curve is 0.74 which means that our model is pretty efficient in separating the not pizza from the pizza model. The Confusion Matrix shows that our True Positive(TP), True Negative(TN), False Positive(FP), False Negative(FN) for classifying pizza class from not pizza for our first/original model is 215, 223, 82, 70, respectively.

Confusion matrix and ROC curve are common evaluation tools used in classification tasks in machine learning. Here's why they are useful:

Confusion Matrix:

```
By displaying the numbers of true positive, true negative, false positive, and fals
e negative predictions, the confusion matrix offers a thorough analysis of the mode
l's performance. In order to evaluate the model's performance for both balanced and
unbalanced classes, it is helpful in determining the model's sensitivity (recall),
specificity, precision, and accuracy.
```

ROC Curve (Receiver Operating Characteristic Curve):

```
The ROC helps visualize the trade-off between sensitivity and specificity by provid
ing a graphical depiction of the true positive rate (sensitivity) against the false
positive rate (1-specificity) for various threshold values. Greater values indicate
greater performance, while the Area Under the ROC Curve (AUC-ROC) offers a single
scalar number measuring the model's performance.
```

Understanding how well the model separates the various classes in the dataset is dependent on these assessment methods. They assist in evaluating many models to determine which is most appropriate for the job at hand, offer insights into the model's advantages and disadvantages, and direct parameter adjustment.

Build another base CNN, but at point c add an extra hidden layer with 32 units of conv2d. Repeat all the other steps. What happened to the accuracy of the model? Why?

Then we built another CNN model called "model2" with the additional hidden layer having Conv2d 32 units and we repeated all the other steps as below:

In [ ]:

```
# Initialize the model
model2 = Sequential()
input_shape = (224, 224, 3)

# Input layer
model2.add(Input(shape=input_shape))

# Data augmentation
datagen2 = ImageDataGenerator(
    horizontal_flip=True,
    vertical_flip=True,
    rotation_range=0.2
)

# Two hidden layers (original)
model2.add(Conv2D(16, (3, 3), activation='relu', input_shape=input_shape))
model2.add(MaxPooling2D((2, 2)))
model2.add(BatchNormalization())

model2.add(Conv2D(24, (3, 3), activation='relu'))
model2.add(MaxPooling2D((2, 2)))
model2.add(BatchNormalization())

# Extra hidden layer (32 units of Conv2D)
model2.add(Conv2D(32, (3, 3), activation='relu'))
model2.add(MaxPooling2D((2, 2)))
model2.add(BatchNormalization())

# Flatten layer and a dense layer
model2.add(Flatten())
model2.add(Dense(8, activation='relu'))

# Final classifier (dense layer)
num_classes=1
model2.add(Dense(num_classes, activation='sigmoid'))

# Compile the model
# compile the CNN
model2.compile(
    loss = tf.keras.losses.BinaryCrossentropy(from_logits = True),
    optimizer = tf.keras.optimizers.Adam(learning_rate = 3e-5),
    metrics =["accuracy"],
    )

# Print the model summary
model2.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 222, 222, 16)      448

 max_pooling2d_2 (MaxPoolin  (None, 111, 111, 16)      0
 g2D)

 batch_normalization_2 (Bat  (None, 111, 111, 16)      64
 chNormalization)

 conv2d_3 (Conv2D)           (None, 109, 109, 24)      3480

 max_pooling2d_3 (MaxPoolin  (None, 54, 54, 24)        0
 g2D)

 batch_normalization_3 (Bat  (None, 54, 54, 24)        96
 chNormalization)

 conv2d_4 (Conv2D)           (None, 52, 52, 32)        6944
```

```
 max_pooling2d_4 (MaxPoolin  (None, 26, 26, 32)        0
 g2D)

 batch_normalization_4 (Bat  (None, 26, 26, 32)        128
 chNormalization)

 flatten_1 (Flatten)         (None, 21632)             0

 dense_2 (Dense)             (None, 8)                 173064

 dense_3 (Dense)             (None, 1)                 9

=================================================================
Total params: 184233 (719.66 KB)
Trainable params: 184089 (719.10 KB)
Non-trainable params: 144 (576.00 Byte)
_________________________________________________________________
```

In [ ]:

```
# Data augmentation
datagen2 = ImageDataGenerator(
    horizontal_flip=True,
    vertical_flip=True,
    rotation_range=0.2
)
```

In [ ]:

```
checkpoint_path2 = "training_second_model/cp.ckpt"
checkpoint_dir2 = os.path.dirname(checkpoint_path2)

# Create a callback that saves the model's weights
cp_callback2 = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_path2,
                                                 save_weights_only=True,
                                                 verbose=1)
```

In [ ]:

```
# do it here
batch_size=64
epochs=128
# Train the model
history = model2.fit(
    datagen2.flow(X_train, y_train, batch_size=batch_size),
    epochs=epochs,
    validation_data=(X_test, y_test),
    callbacks=[cp_callback2],
    verbose=1
)
```

```
Epoch 1/128
```

```
C:\Users\Faiza Anan Noor\anaconda3\Lib\site-packages\keras\src\backend.py:5818: UserWarni
ng: "`binary_crossentropy` received `from_logits=True`, but the `output` argument was pro
duced by a Sigmoid activation and thus does not represent logits. Was this intended?
  output, from_logits = _get_logits(
```

```
22/22 [==============================] - ETA: 0s - loss: 0.7011 - accuracy: 0.5974
Epoch 1: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 44s 2s/step - loss: 0.7011 - accuracy: 0.5974 -
val_loss: 0.6884 - val_accuracy: 0.5085
Epoch 2/128
22/22 [==============================] - ETA: 0s - loss: 0.6078 - accuracy: 0.6781
Epoch 2: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.6078 - accuracy: 0.6781 -
val_loss: 0.6984 - val_accuracy: 0.5034
Epoch 3/128
22/22 [==============================] - ETA: 0s - loss: 0.5873 - accuracy: 0.6904
Epoch 3: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.5873 - accuracy: 0.6904 -
```

```
val_loss: 0.6999 - val_accuracy: 0.5034
Epoch 4/128
22/22 [==============================] - ETA: 0s - loss: 0.5710 - accuracy: 0.7180
Epoch 4: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.5710 - accuracy: 0.7180 -
val_loss: 0.7171 - val_accuracy: 0.5034
Epoch 5/128
22/22 [==============================] - ETA: 0s - loss: 0.5433 - accuracy: 0.7326
Epoch 5: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.5433 - accuracy: 0.7326 -
val_loss: 0.7036 - val_accuracy: 0.5034
Epoch 6/128
22/22 [==============================] - ETA: 0s - loss: 0.5309 - accuracy: 0.7478
Epoch 6: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.5309 - accuracy: 0.7478 -
val_loss: 0.7015 - val_accuracy: 0.5085
Epoch 7/128
22/22 [==============================] - ETA: 0s - loss: 0.5223 - accuracy: 0.7580
Epoch 7: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.5223 - accuracy: 0.7580 -
val_loss: 0.7245 - val_accuracy: 0.5068
Epoch 8/128
22/22 [==============================] - ETA: 0s - loss: 0.5035 - accuracy: 0.7645
Epoch 8: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.5035 - accuracy: 0.7645 -
val_loss: 0.6909 - val_accuracy: 0.5373
Epoch 9/128
22/22 [==============================] - ETA: 0s - loss: 0.4874 - accuracy: 0.7573
Epoch 9: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.4874 - accuracy: 0.7573 -
val_loss: 0.6842 - val_accuracy: 0.5542
Epoch 10/128
22/22 [==============================] - ETA: 0s - loss: 0.4906 - accuracy: 0.7718
Epoch 10: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.4906 - accuracy: 0.7718 -
val_loss: 0.6613 - val_accuracy: 0.5881
Epoch 11/128
22/22 [==============================] - ETA: 0s - loss: 0.4656 - accuracy: 0.7863
Epoch 11: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.4656 - accuracy: 0.7863 -
val_loss: 0.6543 - val_accuracy: 0.6051
Epoch 12/128
22/22 [==============================] - ETA: 0s - loss: 0.4578 - accuracy: 0.7951
Epoch 12: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.4578 - accuracy: 0.7951 -
val_loss: 0.6348 - val_accuracy: 0.6186
Epoch 13/128
22/22 [==============================] - ETA: 0s - loss: 0.4505 - accuracy: 0.7900
Epoch 13: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.4505 - accuracy: 0.7900 -
val_loss: 0.6094 - val_accuracy: 0.6627
Epoch 14/128
22/22 [==============================] - ETA: 0s - loss: 0.4467 - accuracy: 0.7987
Epoch 14: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.4467 - accuracy: 0.7987 -
val_loss: 0.5900 - val_accuracy: 0.6780
Epoch 15/128
22/22 [==============================] - ETA: 0s - loss: 0.4323 - accuracy: 0.8016
Epoch 15: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.4323 - accuracy: 0.8016 -
val_loss: 0.5965 - val_accuracy: 0.6661
Epoch 16/128
22/22 [==============================] - ETA: 0s - loss: 0.4267 - accuracy: 0.8045
Epoch 16: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.4267 - accuracy: 0.8045 -
val_loss: 0.5763 - val_accuracy: 0.6814
Epoch 17/128
22/22 [==============================] - ETA: 0s - loss: 0.4240 - accuracy: 0.8190
Epoch 17: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.4240 - accuracy: 0.8190 -
val_loss: 0.5671 - val_accuracy: 0.7017
Epoch 18/128
22/22 [==============================] - ETA: 0s - loss: 0.4016 - accuracy: 0.8219
```

```
22/22 [==============================] - ETA: 0s - loss: 0.4016 - accuracy: 0.8219
Epoch 18: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.4016 - accuracy: 0.8219 -
val_loss: 0.5558 - val_accuracy: 0.7034
Epoch 19/128
22/22 [==============================] - ETA: 0s - loss: 0.4059 - accuracy: 0.8132
Epoch 19: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.4059 - accuracy: 0.8132 -
val_loss: 0.5572 - val_accuracy: 0.7136
Epoch 20/128
22/22 [==============================] - ETA: 0s - loss: 0.3905 - accuracy: 0.8401
Epoch 20: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.3905 - accuracy: 0.8401 -
val_loss: 0.5444 - val_accuracy: 0.7237
Epoch 21/128
22/22 [==============================] - ETA: 0s - loss: 0.3878 - accuracy: 0.8358
Epoch 21: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.3878 - accuracy: 0.8358 -
val_loss: 0.5440 - val_accuracy: 0.7288
Epoch 22/128
22/22 [==============================] - ETA: 0s - loss: 0.3693 - accuracy: 0.8401
Epoch 22: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.3693 - accuracy: 0.8401 -
val_loss: 0.5345 - val_accuracy: 0.7356
Epoch 23/128
22/22 [==============================] - ETA: 0s - loss: 0.3658 - accuracy: 0.8532
Epoch 23: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.3658 - accuracy: 0.8532 -
val_loss: 0.5432 - val_accuracy: 0.7220
Epoch 24/128
22/22 [==============================] - ETA: 0s - loss: 0.3506 - accuracy: 0.8597
Epoch 24: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.3506 - accuracy: 0.8597 -
val_loss: 0.5349 - val_accuracy: 0.7271
Epoch 25/128
22/22 [==============================] - ETA: 0s - loss: 0.3549 - accuracy: 0.8612
Epoch 25: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.3549 - accuracy: 0.8612 -
val_loss: 0.5347 - val_accuracy: 0.7169
Epoch 26/128
22/22 [==============================] - ETA: 0s - loss: 0.3448 - accuracy: 0.8626
Epoch 26: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.3448 - accuracy: 0.8626 -
val_loss: 0.5355 - val_accuracy: 0.7169
Epoch 27/128
22/22 [==============================] - ETA: 0s - loss: 0.3398 - accuracy: 0.8641
Epoch 27: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.3398 - accuracy: 0.8641 -
val_loss: 0.5376 - val_accuracy: 0.7169
Epoch 28/128
22/22 [==============================] - ETA: 0s - loss: 0.3289 - accuracy: 0.8634
Epoch 28: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.3289 - accuracy: 0.8634 -
val_loss: 0.5378 - val_accuracy: 0.7169
Epoch 29/128
22/22 [==============================] - ETA: 0s - loss: 0.3279 - accuracy: 0.8641
Epoch 29: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.3279 - accuracy: 0.8641 -
val_loss: 0.5303 - val_accuracy: 0.7237
Epoch 30/128
22/22 [==============================] - ETA: 0s - loss: 0.3069 - accuracy: 0.8830
Epoch 30: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.3069 - accuracy: 0.8830 -
val_loss: 0.5264 - val_accuracy: 0.7339
Epoch 31/128
22/22 [==============================] - ETA: 0s - loss: 0.3153 - accuracy: 0.8786
Epoch 31: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.3153 - accuracy: 0.8786 -
val_loss: 0.5273 - val_accuracy: 0.7305
Epoch 32/128
22/22 [==============================] - ETA: 0s - loss: 0.3030 - accuracy: 0.8888
Epoch 32: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.3030 - accuracy: 0.8888 -
```

```
22/22 [==============================] - 30s 1s/step - loss: 0.3030 - accuracy: 0.8888 -
val_loss: 0.5260 - val_accuracy: 0.7356
Epoch 33/128
22/22 [==============================] - ETA: 0s - loss: 0.2976 - accuracy: 0.8866
Epoch 33: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.2976 - accuracy: 0.8866 -
val_loss: 0.5279 - val_accuracy: 0.7373
Epoch 34/128
22/22 [==============================] - ETA: 0s - loss: 0.2889 - accuracy: 0.8874
Epoch 34: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.2889 - accuracy: 0.8874 -
val_loss: 0.5284 - val_accuracy: 0.7441
Epoch 35/128
22/22 [==============================] - ETA: 0s - loss: 0.2936 - accuracy: 0.8910
Epoch 35: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.2936 - accuracy: 0.8910 -
val_loss: 0.5244 - val_accuracy: 0.7508
Epoch 36/128
22/22 [==============================] - ETA: 0s - loss: 0.2675 - accuracy: 0.9084
Epoch 36: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.2675 - accuracy: 0.9084 -
val_loss: 0.5312 - val_accuracy: 0.7390
Epoch 37/128
22/22 [==============================] - ETA: 0s - loss: 0.2669 - accuracy: 0.9070
Epoch 37: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.2669 - accuracy: 0.9070 -
val_loss: 0.5219 - val_accuracy: 0.7424
Epoch 38/128
22/22 [==============================] - ETA: 0s - loss: 0.2711 - accuracy: 0.9041
Epoch 38: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.2711 - accuracy: 0.9041 -
val_loss: 0.5170 - val_accuracy: 0.7492
Epoch 39/128
22/22 [==============================] - ETA: 0s - loss: 0.2548 - accuracy: 0.9230
Epoch 39: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 29s 1s/step - loss: 0.2548 - accuracy: 0.9230 -
val_loss: 0.5151 - val_accuracy: 0.7525
Epoch 40/128
22/22 [==============================] - ETA: 0s - loss: 0.2571 - accuracy: 0.9004
Epoch 40: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.2571 - accuracy: 0.9004 -
val_loss: 0.5211 - val_accuracy: 0.7576
Epoch 41/128
22/22 [==============================] - ETA: 0s - loss: 0.2427 - accuracy: 0.9193
Epoch 41: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.2427 - accuracy: 0.9193 -
val_loss: 0.5172 - val_accuracy: 0.7593
Epoch 42/128
22/22 [==============================] - ETA: 0s - loss: 0.2412 - accuracy: 0.9164
Epoch 42: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.2412 - accuracy: 0.9164 -
val_loss: 0.5225 - val_accuracy: 0.7593
Epoch 43/128
22/22 [==============================] - ETA: 0s - loss: 0.2447 - accuracy: 0.9157
Epoch 43: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.2447 - accuracy: 0.9157 -
val_loss: 0.5220 - val_accuracy: 0.7576
Epoch 44/128
22/22 [==============================] - ETA: 0s - loss: 0.2429 - accuracy: 0.9273
Epoch 44: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.2429 - accuracy: 0.9273 -
val_loss: 0.5292 - val_accuracy: 0.7441
Epoch 45/128
22/22 [==============================] - ETA: 0s - loss: 0.2354 - accuracy: 0.9288
Epoch 45: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 36s 1s/step - loss: 0.2354 - accuracy: 0.9288 -
val_loss: 0.5274 - val_accuracy: 0.7576
Epoch 46/128
22/22 [==============================] - ETA: 0s - loss: 0.2343 - accuracy: 0.9251
Epoch 46: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 41s 2s/step - loss: 0.2343 - accuracy: 0.9251 -
val_loss: 0.5293 - val_accuracy: 0.7593
Epoch 47/128
```

```
Epoch 47/128
22/22 [==============================] - ETA: 0s - loss: 0.2214 - accuracy: 0.9310
Epoch 47: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 39s 2s/step - loss: 0.2214 - accuracy: 0.9310 -
val_loss: 0.5280 - val_accuracy: 0.7492
Epoch 48/128
22/22 [==============================] - ETA: 0s - loss: 0.2198 - accuracy: 0.9368
Epoch 48: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 24s 914ms/step - loss: 0.2198 - accuracy: 0.9368
- val_loss: 0.5319 - val_accuracy: 0.7576
Epoch 49/128
22/22 [==============================] - ETA: 0s - loss: 0.2136 - accuracy: 0.9411
Epoch 49: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 20s 822ms/step - loss: 0.2136 - accuracy: 0.9411
- val_loss: 0.5327 - val_accuracy: 0.7508
Epoch 50/128
22/22 [==============================] - ETA: 0s - loss: 0.2087 - accuracy: 0.9411
Epoch 50: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 20s 845ms/step - loss: 0.2087 - accuracy: 0.9411
- val_loss: 0.5373 - val_accuracy: 0.7559
Epoch 51/128
22/22 [==============================] - ETA: 0s - loss: 0.2101 - accuracy: 0.9368
Epoch 51: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 20s 829ms/step - loss: 0.2101 - accuracy: 0.9368
- val_loss: 0.5332 - val_accuracy: 0.7610
Epoch 52/128
22/22 [==============================] - ETA: 0s - loss: 0.2077 - accuracy: 0.9390
Epoch 52: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 28s 1s/step - loss: 0.2077 - accuracy: 0.9390 -
val_loss: 0.5259 - val_accuracy: 0.7593
Epoch 53/128
22/22 [==============================] - ETA: 0s - loss: 0.2074 - accuracy: 0.9339
Epoch 53: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.2074 - accuracy: 0.9339 -
val_loss: 0.5237 - val_accuracy: 0.7627
Epoch 54/128
22/22 [==============================] - ETA: 0s - loss: 0.2025 - accuracy: 0.9390
Epoch 54: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.2025 - accuracy: 0.9390 -
val_loss: 0.5200 - val_accuracy: 0.7661
Epoch 55/128
22/22 [==============================] - ETA: 0s - loss: 0.1968 - accuracy: 0.9477
Epoch 55: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1968 - accuracy: 0.9477 -
val_loss: 0.5223 - val_accuracy: 0.7627
Epoch 56/128
22/22 [==============================] - ETA: 0s - loss: 0.1900 - accuracy: 0.9528
Epoch 56: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.1900 - accuracy: 0.9528 -
val_loss: 0.5193 - val_accuracy: 0.7593
Epoch 57/128
22/22 [==============================] - ETA: 0s - loss: 0.1847 - accuracy: 0.9564
Epoch 57: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1847 - accuracy: 0.9564 -
val_loss: 0.5228 - val_accuracy: 0.7627
Epoch 58/128
22/22 [==============================] - ETA: 0s - loss: 0.1843 - accuracy: 0.9469
Epoch 58: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1843 - accuracy: 0.9469 -
val_loss: 0.5234 - val_accuracy: 0.7627
Epoch 59/128
22/22 [==============================] - ETA: 0s - loss: 0.1785 - accuracy: 0.9600
Epoch 59: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.1785 - accuracy: 0.9600 -
val_loss: 0.5198 - val_accuracy: 0.7593
Epoch 60/128
22/22 [==============================] - ETA: 0s - loss: 0.1782 - accuracy: 0.9571
Epoch 60: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1782 - accuracy: 0.9571 -
val_loss: 0.5238 - val_accuracy: 0.7559
Epoch 61/128
22/22 [==============================] - ETA: 0s - loss: 0.1802 - accuracy: 0.9578
Epoch 61: saving model to training_second_model\cp.ckpt
```

```
Epoch 61: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1802 - accuracy: 0.9578 -
val_loss: 0.5215 - val_accuracy: 0.7644
Epoch 62/128
22/22 [==============================] - ETA: 0s - loss: 0.1661 - accuracy: 0.9644
Epoch 62: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1661 - accuracy: 0.9644 -
val_loss: 0.5224 - val_accuracy: 0.7644
Epoch 63/128
22/22 [==============================] - ETA: 0s - loss: 0.1640 - accuracy: 0.9644
Epoch 63: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1640 - accuracy: 0.9644 -
val_loss: 0.5212 - val_accuracy: 0.7678
Epoch 64/128
22/22 [==============================] - ETA: 0s - loss: 0.1736 - accuracy: 0.9506
Epoch 64: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1736 - accuracy: 0.9506 -
val_loss: 0.5276 - val_accuracy: 0.7712
Epoch 65/128
22/22 [==============================] - ETA: 0s - loss: 0.1568 - accuracy: 0.9629
Epoch 65: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1568 - accuracy: 0.9629 -
val_loss: 0.5248 - val_accuracy: 0.7763
Epoch 66/128
22/22 [==============================] - ETA: 0s - loss: 0.1542 - accuracy: 0.9629
Epoch 66: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1542 - accuracy: 0.9629 -
val_loss: 0.5209 - val_accuracy: 0.7712
Epoch 67/128
22/22 [==============================] - ETA: 0s - loss: 0.1583 - accuracy: 0.9615
Epoch 67: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1583 - accuracy: 0.9615 -
val_loss: 0.5226 - val_accuracy: 0.7797
Epoch 68/128
22/22 [==============================] - ETA: 0s - loss: 0.1555 - accuracy: 0.9600
Epoch 68: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1555 - accuracy: 0.9600 -
val_loss: 0.5271 - val_accuracy: 0.7729
Epoch 69/128
22/22 [==============================] - ETA: 0s - loss: 0.1597 - accuracy: 0.9564
Epoch 69: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.1597 - accuracy: 0.9564 -
val_loss: 0.5276 - val_accuracy: 0.7627
Epoch 70/128
22/22 [==============================] - ETA: 0s - loss: 0.1464 - accuracy: 0.9673
Epoch 70: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.1464 - accuracy: 0.9673 -
val_loss: 0.5242 - val_accuracy: 0.7729
Epoch 71/128
22/22 [==============================] - ETA: 0s - loss: 0.1493 - accuracy: 0.9702
Epoch 71: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1493 - accuracy: 0.9702 -
val_loss: 0.5213 - val_accuracy: 0.7712
Epoch 72/128
22/22 [==============================] - ETA: 0s - loss: 0.1398 - accuracy: 0.9717
Epoch 72: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1398 - accuracy: 0.9717 -
val_loss: 0.5244 - val_accuracy: 0.7678
Epoch 73/128
22/22 [==============================] - ETA: 0s - loss: 0.1358 - accuracy: 0.9651
Epoch 73: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.1358 - accuracy: 0.9651 -
val_loss: 0.5246 - val_accuracy: 0.7780
Epoch 74/128
22/22 [==============================] - ETA: 0s - loss: 0.1408 - accuracy: 0.9731
Epoch 74: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1408 - accuracy: 0.9731 -
val_loss: 0.5297 - val_accuracy: 0.7712
Epoch 75/128
22/22 [==============================] - ETA: 0s - loss: 0.1313 - accuracy: 0.9753
Epoch 75: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1313 - accuracy: 0.9753 -
val_loss: 0.5291 - val_accuracy: 0.7746
```

```
val_loss: 0.5291 - val_accuracy: 0.7746
Epoch 76/128
22/22 [==============================] - ETA: 0s - loss: 0.1326 - accuracy: 0.9731
Epoch 76: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1326 - accuracy: 0.9731 -
val_loss: 0.5249 - val_accuracy: 0.7661
Epoch 77/128
22/22 [==============================] - ETA: 0s - loss: 0.1332 - accuracy: 0.9724
Epoch 77: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.1332 - accuracy: 0.9724 -
val_loss: 0.5274 - val_accuracy: 0.7661
Epoch 78/128
22/22 [==============================] - ETA: 0s - loss: 0.1299 - accuracy: 0.9775
Epoch 78: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1299 - accuracy: 0.9775 -
val_loss: 0.5307 - val_accuracy: 0.7746
Epoch 79/128
22/22 [==============================] - ETA: 0s - loss: 0.1317 - accuracy: 0.9738
Epoch 79: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1317 - accuracy: 0.9738 -
val_loss: 0.5345 - val_accuracy: 0.7746
Epoch 80/128
22/22 [==============================] - ETA: 0s - loss: 0.1205 - accuracy: 0.9840
Epoch 80: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1205 - accuracy: 0.9840 -
val_loss: 0.5273 - val_accuracy: 0.7644
Epoch 81/128
22/22 [==============================] - ETA: 0s - loss: 0.1220 - accuracy: 0.9753
Epoch 81: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1220 - accuracy: 0.9753 -
val_loss: 0.5296 - val_accuracy: 0.7780
Epoch 82/128
22/22 [==============================] - ETA: 0s - loss: 0.1244 - accuracy: 0.9760
Epoch 82: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1244 - accuracy: 0.9760 -
val_loss: 0.5231 - val_accuracy: 0.7763
Epoch 83/128
22/22 [==============================] - ETA: 0s - loss: 0.1216 - accuracy: 0.9767
Epoch 83: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1216 - accuracy: 0.9767 -
val_loss: 0.5347 - val_accuracy: 0.7644
Epoch 84/128
22/22 [==============================] - ETA: 0s - loss: 0.1148 - accuracy: 0.9818
Epoch 84: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1148 - accuracy: 0.9818 -
val_loss: 0.5324 - val_accuracy: 0.7610
Epoch 85/128
22/22 [==============================] - ETA: 0s - loss: 0.1208 - accuracy: 0.9789
Epoch 85: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1208 - accuracy: 0.9789 -
val_loss: 0.5340 - val_accuracy: 0.7695
Epoch 86/128
22/22 [==============================] - ETA: 0s - loss: 0.1170 - accuracy: 0.9789
Epoch 86: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1170 - accuracy: 0.9789 -
val_loss: 0.5303 - val_accuracy: 0.7780
Epoch 87/128
22/22 [==============================] - ETA: 0s - loss: 0.1079 - accuracy: 0.9833
Epoch 87: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1079 - accuracy: 0.9833 -
val_loss: 0.5336 - val_accuracy: 0.7729
Epoch 88/128
22/22 [==============================] - ETA: 0s - loss: 0.1074 - accuracy: 0.9826
Epoch 88: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1074 - accuracy: 0.9826 -
val_loss: 0.5320 - val_accuracy: 0.7780
Epoch 89/128
22/22 [==============================] - ETA: 0s - loss: 0.1105 - accuracy: 0.9782
Epoch 89: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1105 - accuracy: 0.9782 -
val_loss: 0.5364 - val_accuracy: 0.7593
Epoch 90/128
22/22 [==============================] - ETA: 0s - loss: 0.1023 - accuracy: 0.9840
```

```
22/22 [==============================] - ETA: 0s - loss: 0.1023 - accuracy: 0.9840
Epoch 90: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1023 - accuracy: 0.9840 -
val_loss: 0.5330 - val_accuracy: 0.7797
Epoch 91/128
22/22 [==============================] - ETA: 0s - loss: 0.1007 - accuracy: 0.9847
Epoch 91: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1007 - accuracy: 0.9847 -
val_loss: 0.5347 - val_accuracy: 0.7729
Epoch 92/128
22/22 [==============================] - ETA: 0s - loss: 0.1002 - accuracy: 0.9862
Epoch 92: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.1002 - accuracy: 0.9862 -
val_loss: 0.5404 - val_accuracy: 0.7610
Epoch 93/128
22/22 [==============================] - ETA: 0s - loss: 0.1112 - accuracy: 0.9797
Epoch 93: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.1112 - accuracy: 0.9797 -
val_loss: 0.5356 - val_accuracy: 0.7644
Epoch 94/128
22/22 [==============================] - ETA: 0s - loss: 0.0989 - accuracy: 0.9826
Epoch 94: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0989 - accuracy: 0.9826 -
val_loss: 0.5325 - val_accuracy: 0.7780
Epoch 95/128
22/22 [==============================] - ETA: 0s - loss: 0.0992 - accuracy: 0.9826
Epoch 95: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0992 - accuracy: 0.9826 -
val_loss: 0.5325 - val_accuracy: 0.7797
Epoch 96/128
22/22 [==============================] - ETA: 0s - loss: 0.0979 - accuracy: 0.9855
Epoch 96: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0979 - accuracy: 0.9855 -
val_loss: 0.5369 - val_accuracy: 0.7593
Epoch 97/128
22/22 [==============================] - ETA: 0s - loss: 0.0954 - accuracy: 0.9876
Epoch 97: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0954 - accuracy: 0.9876 -
val_loss: 0.5394 - val_accuracy: 0.7729
Epoch 98/128
22/22 [==============================] - ETA: 0s - loss: 0.0947 - accuracy: 0.9862
Epoch 98: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0947 - accuracy: 0.9862 -
val_loss: 0.5419 - val_accuracy: 0.7712
Epoch 99/128
22/22 [==============================] - ETA: 0s - loss: 0.0953 - accuracy: 0.9826
Epoch 99: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0953 - accuracy: 0.9826 -
val_loss: 0.5402 - val_accuracy: 0.7712
Epoch 100/128
22/22 [==============================] - ETA: 0s - loss: 0.0982 - accuracy: 0.9840
Epoch 100: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0982 - accuracy: 0.9840 -
val_loss: 0.5480 - val_accuracy: 0.7593
Epoch 101/128
22/22 [==============================] - ETA: 0s - loss: 0.0931 - accuracy: 0.9818
Epoch 101: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.0931 - accuracy: 0.9818 -
val_loss: 0.5419 - val_accuracy: 0.7746
Epoch 102/128
22/22 [==============================] - ETA: 0s - loss: 0.0882 - accuracy: 0.9884
Epoch 102: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0882 - accuracy: 0.9884 -
val_loss: 0.5453 - val_accuracy: 0.7695
Epoch 103/128
22/22 [==============================] - ETA: 0s - loss: 0.0896 - accuracy: 0.9862
Epoch 103: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0896 - accuracy: 0.9862 -
val_loss: 0.5475 - val_accuracy: 0.7661
Epoch 104/128
22/22 [==============================] - ETA: 0s - loss: 0.0857 - accuracy: 0.9898
Epoch 104: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0857 - accuracy: 0.9898 -
```

```
22/22 [==============================] - 31s 1s/step - loss: 0.0857 - accuracy: 0.9898 -
val_loss: 0.5540 - val_accuracy: 0.7678
Epoch 105/128
22/22 [==============================] - ETA: 0s - loss: 0.0839 - accuracy: 0.9884
Epoch 105: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.0839 - accuracy: 0.9884 -
val_loss: 0.5435 - val_accuracy: 0.7712
Epoch 106/128
22/22 [==============================] - ETA: 0s - loss: 0.0827 - accuracy: 0.9898
Epoch 106: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.0827 - accuracy: 0.9898 -
val_loss: 0.5517 - val_accuracy: 0.7644
Epoch 107/128
22/22 [==============================] - ETA: 0s - loss: 0.0783 - accuracy: 0.9927
Epoch 107: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0783 - accuracy: 0.9927 -
val_loss: 0.5435 - val_accuracy: 0.7780
Epoch 108/128
22/22 [==============================] - ETA: 0s - loss: 0.0761 - accuracy: 0.9927
Epoch 108: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0761 - accuracy: 0.9927 -
val_loss: 0.5441 - val_accuracy: 0.7729
Epoch 109/128
22/22 [==============================] - ETA: 0s - loss: 0.0759 - accuracy: 0.9949
Epoch 109: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0759 - accuracy: 0.9949 -
val_loss: 0.5464 - val_accuracy: 0.7729
Epoch 110/128
22/22 [==============================] - ETA: 0s - loss: 0.0723 - accuracy: 0.9920
Epoch 110: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 30s 1s/step - loss: 0.0723 - accuracy: 0.9920 -
val_loss: 0.5502 - val_accuracy: 0.7746
Epoch 111/128
22/22 [==============================] - ETA: 0s - loss: 0.0725 - accuracy: 0.9935
Epoch 111: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.0725 - accuracy: 0.9935 -
val_loss: 0.5510 - val_accuracy: 0.7746
Epoch 112/128
22/22 [==============================] - ETA: 0s - loss: 0.0737 - accuracy: 0.9884
Epoch 112: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0737 - accuracy: 0.9884 -
val_loss: 0.5583 - val_accuracy: 0.7661
Epoch 113/128
22/22 [==============================] - ETA: 0s - loss: 0.0713 - accuracy: 0.9913
Epoch 113: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0713 - accuracy: 0.9913 -
val_loss: 0.5497 - val_accuracy: 0.7763
Epoch 114/128
22/22 [==============================] - ETA: 0s - loss: 0.0761 - accuracy: 0.9920
Epoch 114: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0761 - accuracy: 0.9920 -
val_loss: 0.5655 - val_accuracy: 0.7695
Epoch 115/128
22/22 [==============================] - ETA: 0s - loss: 0.0705 - accuracy: 0.9891
Epoch 115: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0705 - accuracy: 0.9891 -
val_loss: 0.5434 - val_accuracy: 0.7746
Epoch 116/128
22/22 [==============================] - ETA: 0s - loss: 0.0698 - accuracy: 0.9935
Epoch 116: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0698 - accuracy: 0.9935 -
val_loss: 0.5531 - val_accuracy: 0.7797
Epoch 117/128
22/22 [==============================] - ETA: 0s - loss: 0.0699 - accuracy: 0.9913
Epoch 117: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0699 - accuracy: 0.9913 -
val_loss: 0.5454 - val_accuracy: 0.7831
Epoch 118/128
22/22 [==============================] - ETA: 0s - loss: 0.0662 - accuracy: 0.9964
Epoch 118: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0662 - accuracy: 0.9964 -
val_loss: 0.5449 - val_accuracy: 0.7831
Epoch 119/128
```

```
Epoch 119/128
22/22 [==============================] - ETA: 0s - loss: 0.0695 - accuracy: 0.9913
Epoch 119: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 31s 1s/step - loss: 0.0695 - accuracy: 0.9913 -
val_loss: 0.5451 - val_accuracy: 0.7797
Epoch 120/128
22/22 [==============================] - ETA: 0s - loss: 0.0630 - accuracy: 0.9956
Epoch 120: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.0630 - accuracy: 0.9956 -
val_loss: 0.5450 - val_accuracy: 0.7847
Epoch 121/128
22/22 [==============================] - ETA: 0s - loss: 0.0664 - accuracy: 0.9956
Epoch 121: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.0664 - accuracy: 0.9956 -
val_loss: 0.5524 - val_accuracy: 0.7814
Epoch 122/128
22/22 [==============================] - ETA: 0s - loss: 0.0637 - accuracy: 0.9935
Epoch 122: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 33s 1s/step - loss: 0.0637 - accuracy: 0.9935 -
val_loss: 0.5511 - val_accuracy: 0.7763
Epoch 123/128
22/22 [==============================] - ETA: 0s - loss: 0.0621 - accuracy: 0.9942
Epoch 123: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 35s 1s/step - loss: 0.0621 - accuracy: 0.9942 -
val_loss: 0.5504 - val_accuracy: 0.7831
Epoch 124/128
22/22 [==============================] - ETA: 0s - loss: 0.0606 - accuracy: 0.9956
Epoch 124: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.0606 - accuracy: 0.9956 -
val_loss: 0.5554 - val_accuracy: 0.7780
Epoch 125/128
22/22 [==============================] - ETA: 0s - loss: 0.0591 - accuracy: 0.9949
Epoch 125: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.0591 - accuracy: 0.9949 -
val_loss: 0.5546 - val_accuracy: 0.7763
Epoch 126/128
22/22 [==============================] - ETA: 0s - loss: 0.0575 - accuracy: 0.9964
Epoch 126: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 32s 1s/step - loss: 0.0575 - accuracy: 0.9964 -
val_loss: 0.5606 - val_accuracy: 0.7797
Epoch 127/128
22/22 [==============================] - ETA: 0s - loss: 0.0586 - accuracy: 0.9942
Epoch 127: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 33s 1s/step - loss: 0.0586 - accuracy: 0.9942 -
val_loss: 0.5568 - val_accuracy: 0.7881
Epoch 128/128
22/22 [==============================] - ETA: 0s - loss: 0.0554 - accuracy: 0.9964
Epoch 128: saving model to training_second_model\cp.ckpt
22/22 [==============================] - 33s 1s/step - loss: 0.0554 - accuracy: 0.9964 -
val_loss: 0.5509 - val_accuracy: 0.7864
```

In [ ]:

```
model2.save('Second_model.keras')
```

In [ ]:

```
model2= tf.keras.models.load_model('Second_model.keras')
```

Evaluate the model and report the accuracy

I evaluated the model using model.evaluate passing in our test labels and targets. And then from that variable containing the evaluation results, we take the index 1 to get the accuracy of our model on the test set.

In [ ]:

```
eval_result2 = model2.evaluate(X_test, y_test, verbose=1)

# Print the accuracy
accuracy2 = eval_result2[1] * 100  # Accuracy is typically the second element in the eval
```

```
uation result
print(f'Test Accuracy: {accuracy2:.2f}%')#
```

```
19/19 [==============================] - 8s 42ms/step - loss: 0.5509 - accuracy: 0.7864
Test Accuracy: 78.64%
```

For our second model, the accuracy is 78.64% which seems to be higher than our first model. This is because of the addition of the extra hidden layer.

Make prediction with the test set and use a threshold of 0.5 as boundaries decision between the classes.

Then we make prediction using the test set(X_test) and the variable "predictions" contains the results. Then we set threshold as 0.5 and make binary predictions variable contain predictions as 1 for values that are above 0.5 and 0 otherwise. And this is used using the "predictions" variable defined above containing the real values for the predictions.

In [ ]:

```
threshold=0.5
predictions2=model2.predict(X_test)
binary_labels2 = np.argmax(predictions2, axis=1)
binary_predictions2 = (predictions2 > threshold).astype(int)
#binary_predictions2
```

```
19/19 [==============================] - 8s 43ms/step
```

In [ ]:

```
predictions2[:9]
```

Out[ ]:

```
array([[5.7269961e-01],
       [6.5179271e-01],
       [7.8221798e-05],
       [7.9324299e-01],
       [8.3395606e-01],
       [8.5443377e-01],
       [1.6503903e-01],
       [1.2467336e-02],
       [7.1737725e-01]], dtype=float32)
```

show metrics like confusion matrix or ROC curve or both (sklearn has already implemented all these stuff)

In [ ]:

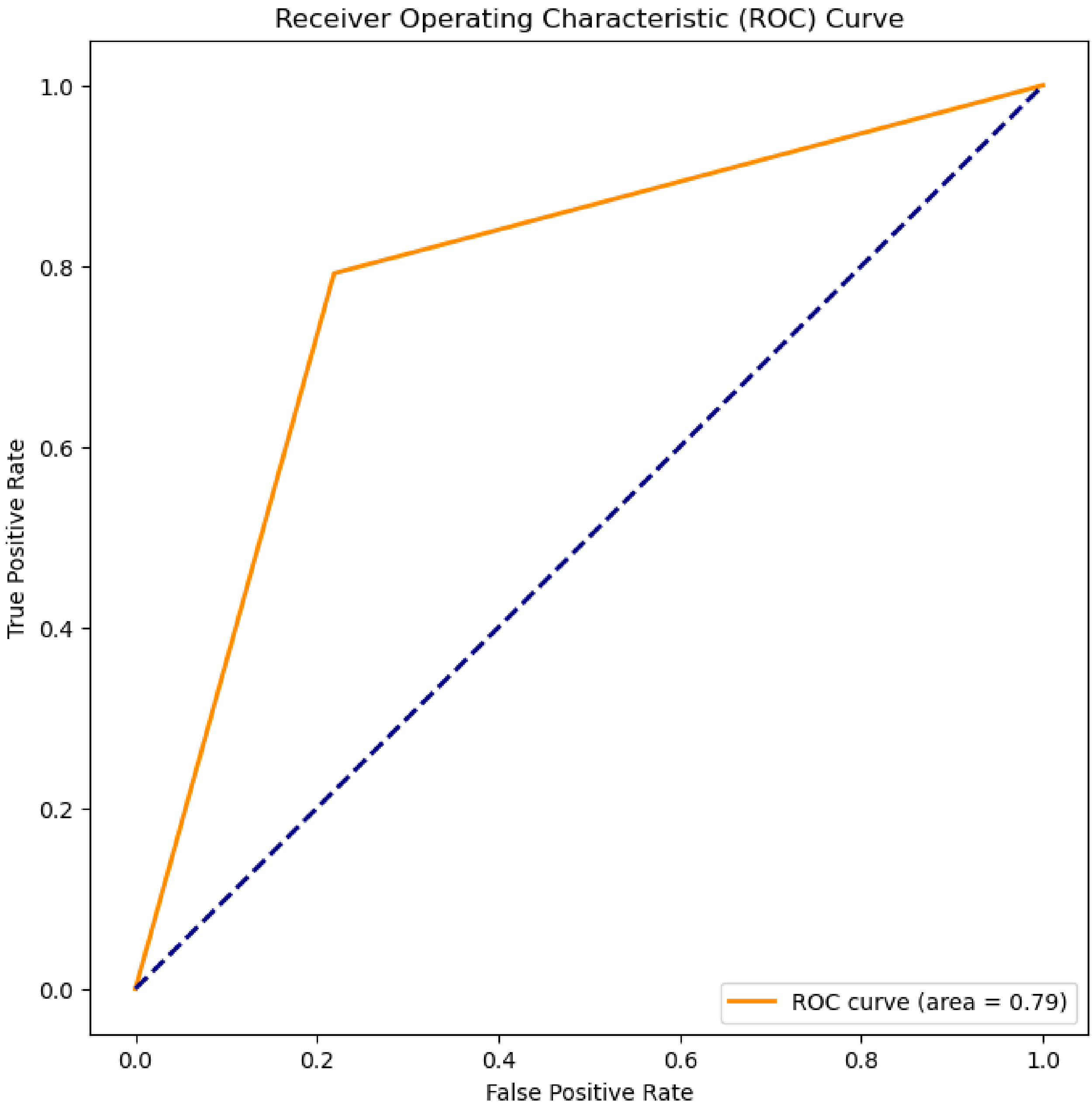
```
print("Confusion Matrix and ROC curve of model 2 is given below : ")
```

```
Confusion Matrix and ROC curve of model 2 is given below :
```

In [ ]:

```
# Plot ROC Curve
fpr, tpr, thresholds = roc_curve(y_test,  binary_predictions2.flatten())
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 8))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
```

```
plt.show()
```

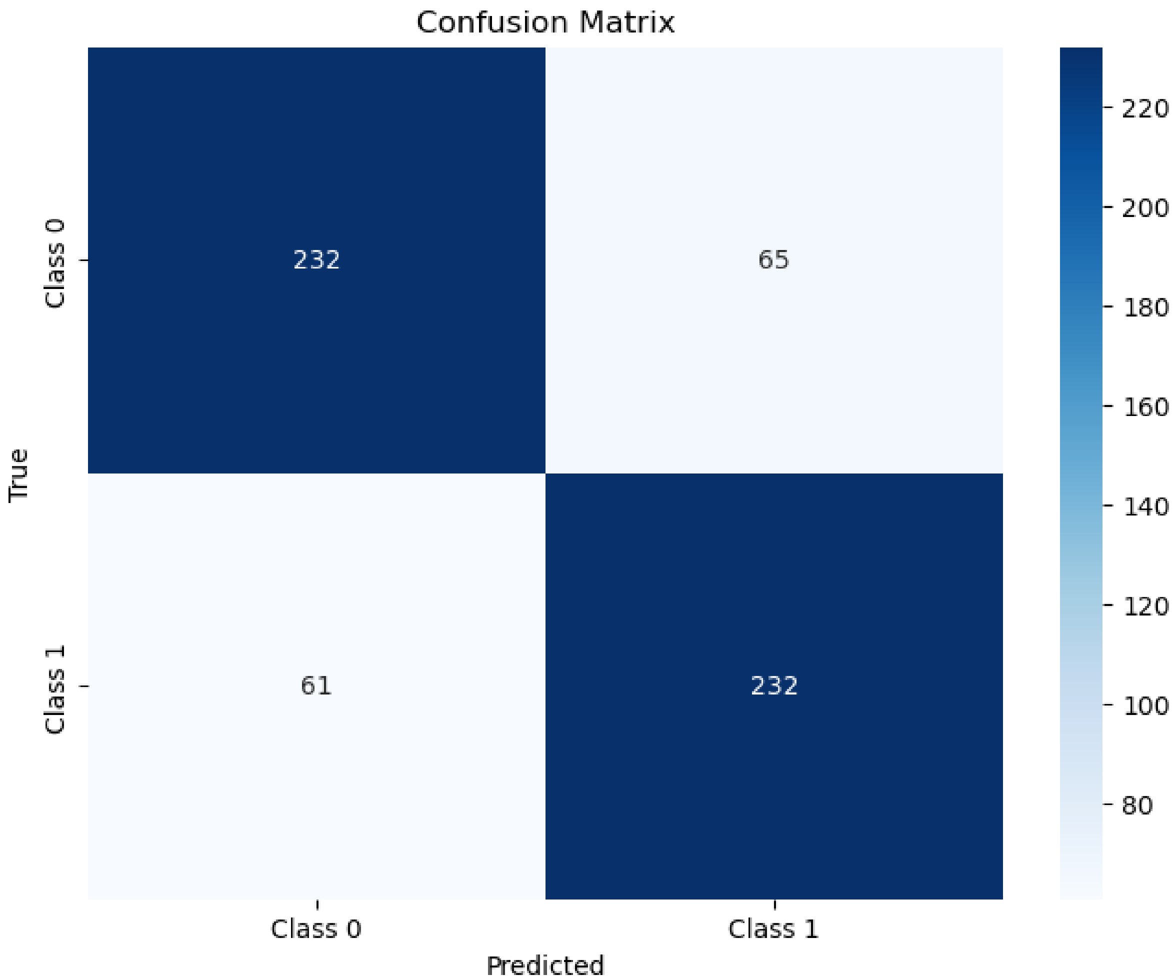


In [ ]:

```
# Calculate and display the confusion matrix
conf_matrix = confusion_matrix(y_test,  binary_predictions2.flatten())
print("Confusion Matrix:")
print(conf_matrix)

# Plot Confusion Matrix using Matplotlib
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Class 0', 'Cla
ss 1'], yticklabels=['Class 0', 'Class 1'])
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()
```

```
Confusion Matrix:
[[232  65]
 [ 61 232]]
```

[61 232]]



For our second model, the area under the curve is 0.79 which means that our model is pretty efficient in separating the not pizza from the pizza model and is much better than the first model. Addition of the extra hidden layer seemed to have made it more efficient in distinguishing between the two classes. The Confusion Matrix shows that our True Positive(TP), True Negative(TN), False Positive(FP), False Negative(FN) for classifying pizza class from not pizza for our first/original model is 232, 232, 65, 61, respectively. These results also mean that the model's performance is much better than the first model because the wrongly misinterpretted classes are much less in number and the right ones are more in number.

show predictions

Then we tried to show some predictions by plotting a test image against our predicted Label Results. We can see that almost 78% of the time, it gets correct results.

In [ ]:

```
def show_some_prediction(number_of_subplot, test_set, predictions, name_of_the_labels):
    for i in range(number_of_subplot):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(test_set[i])
        plt.title(f'{name_of_the_labels[predictions[i]]}')
        plt.axis("off")
    plt.show()
show_some_prediction(9, X_test, binary_predictions2.flatten(), label_names)
```

not_pizza

not_pizza

pizza

In [ ]:

```

```

In [ ]:

```

```