In [ ]:
```python
# If running in a Jupyter notebook, use this magic command to install TensorFlow
# !pip install tensorflow

# Step 1: Import necessary libraries
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
import numpy as np
import os
from sklearn.model_selection import train_test_split
import shutil

# Step 2: Create train and validation directories within the current directory
original_data_dir_pizza = 'pizza'
original_data_dir_not_pizza = 'not_pizza'
base_dir = '.'

train_dir = os.path.join(base_dir, 'train')
validation_dir = os.path.join(base_dir, 'validation')

train_pizza_dir = os.path.join(train_dir, 'pizza')
validation_pizza_dir = os.path.join(validation_dir, 'pizza')
train_not_pizza_dir = os.path.join(train_dir, 'not_pizza')
validation_not_pizza_dir = os.path.join(validation_dir, 'not_pizza')

# Create directories
os.makedirs(train_pizza_dir, exist_ok=True)
os.makedirs(validation_pizza_dir, exist_ok=True)
os.makedirs(train_not_pizza_dir, exist_ok=True)
os.makedirs(validation_not_pizza_dir, exist_ok=True)

# Step 3: Split the data into training and validation sets
pizza_images = os.listdir(original_data_dir_pizza)
not_pizza_images = os.listdir(original_data_dir_not_pizza)

train_pizza, val_pizza = train_test_split(pizza_images, test_size=0.2, random_state
train_not_pizza, val_not_pizza = train_test_split(not_pizza_images, test_size=0.2,

# Copy images to train and validation directories
def copy_images(image_list, source_dir, target_dir):
    for image in image_list:
        shutil.copy(os.path.join(source_dir, image), target_dir)

copy_images(train_pizza, original_data_dir_pizza, train_pizza_dir)
copy_images(val_pizza, original_data_dir_pizza, validation_pizza_dir)
copy_images(train_not_pizza, original_data_dir_not_pizza, train_not_pizza_dir)
copy_images(val_not_pizza, original_data_dir_not_pizza, validation_not_pizza_dir)

# Step 4: Data preprocessing
# Image Data Generator for training set with augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True
)

# Image Data Generator for validation set without augmentation
validation_datagen = ImageDataGenerator(rescale=1./255)

# Training set
training_set = train_datagen.flow_from_directory(
    train_dir,
    target_size=(64, 64),
    batch_size=32,
    class_mode='binary'
)

# Validation set
validation_set = validation_datagen.flow_from_directory(
    validation_dir,
    target_size=(64, 64),
    batch_size=32,
    class_mode='binary'
)

# Step 5: Build the CNN model
cnn = tf.keras.models.Sequential()

# Convolution
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu', input_
# Pooling
cnn.add(tf.keras.layers.MaxPooling2D(pool_size=2, strides=2))

# Adding a second convolutional layer
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPooling2D(pool_size=2, strides=2))

# Flattening
cnn.add(tf.keras.layers.Flatten())

# Full connection
cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))
cnn.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))

# Step 6: Compile the CNN
cnn.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Step 7: Train the CNN
cnn.fit(x=training_set, validation_data=validation_set, epochs=25)

# Step 8: Make predictions on new images
def predict_image(image_path):
    test_image = image.load_img(image_path, target_size=(64, 64))
    test_image = image.img_to_array(test_image)
    test_image = np.expand_dims(test_image, axis=0)
    result = cnn.predict(test_image)
    if result[0][0] == 1:
        prediction = 'not pizza'
    else:
```

```python
        prediction = 'pizza'
    return prediction

# Example usage
image1_path = 'some_image.jpg' # Replace with your image path
print(f'The image is {predict_image(image1_path)}')
```

```
Found 1572 images belonging to 2 classes.
Found 394 images belonging to 2 classes.
```

```
C:\Users\joelr\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfr
a8p0\LocalCache\local-packages\Python311\site-packages\keras\src\layers\convolutiona
l\base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument t
o a layer. When using Sequential models, prefer using an `Input(shape)` object as th
e first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
Epoch 1/25
```

```
C:\Users\joelr\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfr
a8p0\LocalCache\local-packages\Python311\site-packages\keras\src\trainers\data_adapt
ers\py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class should call `supe
r().__init__(**kwargs)` in its constructor. `**kwargs` can include `workers`, `use_m
ultiprocessing`, `max_queue_size`. Do not pass these arguments to `fit()`, as they w
ill be ignored.
  self._warn_if_super_not_called()
```

```
50/50 ━━━━━━━━━━━━━━━━━━━━ 9s 142ms/step - accuracy: 0.5306 - loss: 0.7376 - val_acc
uracy: 0.5990 - val_loss: 0.6763
Epoch 2/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 46ms/step - accuracy: 0.6314 - loss: 0.6456 - val_accu
racy: 0.6853 - val_loss: 0.6245
Epoch 3/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 45ms/step - accuracy: 0.6782 - loss: 0.6084 - val_accu
racy: 0.6421 - val_loss: 0.6117
Epoch 4/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 45ms/step - accuracy: 0.7345 - loss: 0.5622 - val_accu
racy: 0.6624 - val_loss: 0.6494
Epoch 5/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 49ms/step - accuracy: 0.6991 - loss: 0.5779 - val_accu
racy: 0.6980 - val_loss: 0.6219
Epoch 6/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 48ms/step - accuracy: 0.7452 - loss: 0.5427 - val_accu
racy: 0.7234 - val_loss: 0.5460
Epoch 7/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 46ms/step - accuracy: 0.7533 - loss: 0.5042 - val_accu
racy: 0.7716 - val_loss: 0.5299
Epoch 8/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 46ms/step - accuracy: 0.7891 - loss: 0.4827 - val_accu
racy: 0.7640 - val_loss: 0.5038
Epoch 9/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 46ms/step - accuracy: 0.7701 - loss: 0.4811 - val_accu
racy: 0.7513 - val_loss: 0.5170
Epoch 10/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 48ms/step - accuracy: 0.7932 - loss: 0.4630 - val_accu
racy: 0.7513 - val_loss: 0.5339
Epoch 11/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 48ms/step - accuracy: 0.7979 - loss: 0.4368 - val_accu
racy: 0.7589 - val_loss: 0.5278
Epoch 12/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 49ms/step - accuracy: 0.8116 - loss: 0.4298 - val_accu
racy: 0.7462 - val_loss: 0.5161
Epoch 13/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 48ms/step - accuracy: 0.8221 - loss: 0.4145 - val_accu
racy: 0.7487 - val_loss: 0.5243
Epoch 14/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 49ms/step - accuracy: 0.8087 - loss: 0.4021 - val_accu
racy: 0.7716 - val_loss: 0.5313
Epoch 15/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 49ms/step - accuracy: 0.8318 - loss: 0.3864 - val_accu
racy: 0.7843 - val_loss: 0.5114
Epoch 16/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 49ms/step - accuracy: 0.8373 - loss: 0.3737 - val_accu
racy: 0.7766 - val_loss: 0.5192
Epoch 17/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 49ms/step - accuracy: 0.8037 - loss: 0.4097 - val_accu
racy: 0.7766 - val_loss: 0.5013
Epoch 18/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 48ms/step - accuracy: 0.8509 - loss: 0.3534 - val_accu
racy: 0.7716 - val_loss: 0.4879
Epoch 19/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 49ms/step - accuracy: 0.8692 - loss: 0.3460 - val_accu
racy: 0.7792 - val_loss: 0.4976
```

```
Epoch 20/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 47ms/step - accuracy: 0.8574 - loss: 0.3291 - val_accu
racy: 0.7690 - val_loss: 0.5707
Epoch 21/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 47ms/step - accuracy: 0.8533 - loss: 0.3310 - val_accu
racy: 0.7741 - val_loss: 0.6037
Epoch 22/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 48ms/step - accuracy: 0.8739 - loss: 0.3215 - val_accu
racy: 0.7944 - val_loss: 0.5209
Epoch 23/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 47ms/step - accuracy: 0.8584 - loss: 0.3266 - val_accu
racy: 0.7817 - val_loss: 0.5011
Epoch 24/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 48ms/step - accuracy: 0.8637 - loss: 0.3025 - val_accu
racy: 0.7843 - val_loss: 0.5194
Epoch 25/25
50/50 ━━━━━━━━━━━━━━━━━━━━ 3s 48ms/step - accuracy: 0.8673 - loss: 0.3059 - val_accu
racy: 0.7843 - val_loss: 0.4945
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
The image is not pizza
```

```
In [ ]:
```