```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tqdm import tqdm
import cv2

import warnings
warnings.filterwarnings('ignore')
import os

from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Activation, Dropout,
Flatten, Dense, BatchNormalization
from keras.preprocessing.image import ImageDataGenerator,
img_to_array, load_img
from keras.utils.vis_utils import plot_model
from glob import glob
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

train_path = "/kaggle/input/waste-classification-data/DATASET/TRAIN/"
test_path = "/kaggle/input/waste-classification-data/DATASET/TEST/"

model = Sequential()
model.add(Conv2D(32,(3,3),input_shape = (224,224,3)))
model.add(Activation("relu"))
model.add(MaxPooling2D())

model.add(Conv2D(64,(3,3)))
model.add(Activation("relu"))
model.add(MaxPooling2D())

model.add(Conv2D(128,(3,3)))
model.add(Activation("relu"))
model.add(MaxPooling2D())

model.add(Flatten())
model.add(Dense(256))
model.add(Activation("relu"))
model.add(Dropout(0.5))
model.add(Dense(64))
model.add(Activation("relu"))
model.add(Dropout(0.5))
model.add(Dense(2))
model.add(Activation("sigmoid"))

model.compile(loss = "binary_crossentropy",
              optimizer = "adam",
```

```
              metrics = ["accuracy"])

batch_size = 256

train_datagen = ImageDataGenerator(rescale= 1./255)

test_datagen = ImageDataGenerator(rescale= 1./255)

train_generator = train_datagen.flow_from_directory(
        train_path,
        target_size= (224,224),
        batch_size = batch_size,
        color_mode= "rgb",
        class_mode= "categorical")

test_generator = test_datagen.flow_from_directory(
        test_path,
        target_size= (224,224),
        batch_size = batch_size,
        color_mode= "rgb",
        class_mode= "categorical")
```

```
Found 22564 images belonging to 2 classes.
Found 2513 images belonging to 2 classes.
```

```
hist = model.fit_generator(
        generator = train_generator,
        epochs=10,
        validation_data = test_generator)
```

```
Epoch 1/10
89/89 [==============================] - 1205s 13s/step - loss: 0.5438
- accuracy: 0.7407 - val_loss: 0.2958 - val_accuracy: 0.8830
Epoch 2/10
89/89 [==============================] - 1187s 13s/step - loss: 0.3958
- accuracy: 0.8368 - val_loss: 0.3822 - val_accuracy: 0.8297
Epoch 3/10
89/89 [==============================] - 1188s 13s/step - loss: 0.3622
- accuracy: 0.8540 - val_loss: 0.2889 - val_accuracy: 0.8838
Epoch 4/10
89/89 [==============================] - 1188s 13s/step - loss: 0.3216
- accuracy: 0.8729 - val_loss: 0.2690 - val_accuracy: 0.8961
Epoch 5/10
89/89 [==============================] - 1187s 13s/step - loss: 0.3085
- accuracy: 0.8790 - val_loss: 0.2681 - val_accuracy: 0.8922
Epoch 6/10
89/89 [==============================] - 1189s 13s/step - loss: 0.2774
- accuracy: 0.8884 - val_loss: 0.3790 - val_accuracy: 0.8496
Epoch 7/10
89/89 [==============================] - 1188s 13s/step - loss: 0.2575
- accuracy: 0.8950 - val_loss: 0.2985 - val_accuracy: 0.8953
```

```
Epoch 8/10
89/89 [==============================] - 1190s 13s/step - loss: 0.2154
- accuracy: 0.9174 - val_loss: 0.2729 - val_accuracy: 0.9109
Epoch 9/10
89/89 [==============================] - 1187s 13s/step - loss: 0.1716
- accuracy: 0.9358 - val_loss: 0.2889 - val_accuracy: 0.8949
Epoch 10/10
89/89 [==============================] - 1188s 13s/step - loss: 0.1535
- accuracy: 0.9423 - val_loss: 0.3819 - val_accuracy: 0.8747

Y_pred = model.predict(test_generator)
y_pred = np.argmax(Y_pred, axis=1)

y_true = test_generator.classes
conf_mat = confusion_matrix(y_true, y_pred)

accuracy = model.evaluate_generator(test_generator)
print("Accuracy:", accuracy)

Accuracy: [0.3819483518600464, 0.8746517896652222]
```