```
import pandas as pd
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense
from google.colab import drive

drive.mount="/content/drive/MyDrive/COMP1801_CourseworkDataset2_images
"

# Load the CSV file
csv_file_path =
"/content/drive/MyDrive/COMP1801_CourseworkDataset2_images/COMP1801_Co
urseworkDataset2_images_metadata.csv"
df = pd.read_csv(csv_file_path)

# Define image directory
image_dir =
"/content/drive/MyDrive/COMP1801_CourseworkDataset2_images"

# Split the data into training and testing sets
train_df, test_df = train_test_split(df, test_size=0.2,
random_state=42)

# Create an ImageDataGenerator for data augmentation
datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2,
zoom_range=0.2, horizontal_flip=True)

# Create data generators
train_generator = datagen.flow_from_dataframe(
    dataframe=train_df,
    directory=image_dir,
    x_col="Image Filename",
    y_col="Defect",
    target_size=(128, 128),  # Adjust the target size based on your
images
    batch_size=32,
    class_mode="binary"  # Change to "categorical" if you have
multiple classes
)

test_generator = datagen.flow_from_dataframe(
    dataframe=test_df,
    directory=image_dir,
    x_col="Image Filename",
    y_col="Defect",
    target_size=(128, 128),
    batch_size=32,
    class_mode="binary"
```

```
)

# Build a simple CNN model
model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(128, 128, 3),
activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(1, activation='sigmoid'))  # Binary classification,
change for multiclass

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

# Train the model
model.fit(train_generator, epochs=10, validation_data=test_generator)

# Evaluate the model
accuracy = model.evaluate(test_generator)[1]
print(f"Test Accuracy: {accuracy}")
```

```
Found 800 validated image filenames belonging to 2 classes.
Found 200 validated image filenames belonging to 2 classes.
Epoch 1/10
25/25 [==============================] - 25s 911ms/step - loss: 2.0372
- accuracy: 0.5337 - val_loss: 0.6522 - val_accuracy: 0.6150
Epoch 2/10
25/25 [==============================] - 23s 944ms/step - loss: 0.6350
- accuracy: 0.6275 - val_loss: 0.6457 - val_accuracy: 0.7950
Epoch 3/10
25/25 [==============================] - 22s 875ms/step - loss: 0.5869
- accuracy: 0.7075 - val_loss: 0.5036 - val_accuracy: 0.8150
Epoch 4/10
25/25 [==============================] - 23s 912ms/step - loss: 0.3999
- accuracy: 0.8512 - val_loss: 0.2965 - val_accuracy: 0.9050
Epoch 5/10
25/25 [==============================] - 22s 868ms/step - loss: 0.2304
- accuracy: 0.9300 - val_loss: 0.2151 - val_accuracy: 0.9450
Epoch 6/10
25/25 [==============================] - 22s 883ms/step - loss: 0.1718
- accuracy: 0.9538 - val_loss: 0.1873 - val_accuracy: 0.9350
Epoch 7/10
25/25 [==============================] - 29s 1s/step - loss: 0.1535 -
accuracy: 0.9538 - val_loss: 0.1577 - val_accuracy: 0.9500
Epoch 8/10
25/25 [==============================] - 23s 946ms/step - loss: 0.1497
- accuracy: 0.9563 - val_loss: 0.1371 - val_accuracy: 0.9650
Epoch 9/10
```

```
25/25 [==============================] - 27s 1s/step - loss: 0.1507 -
accuracy: 0.9488 - val_loss: 0.2246 - val_accuracy: 0.9100
Epoch 10/10
25/25 [==============================] - 24s 963ms/step - loss: 0.1454
- accuracy: 0.9625 - val_loss: 0.1545 - val_accuracy: 0.9550
7/7 [==============================] - 2s 286ms/step - loss: 0.1392 -
accuracy: 0.9600
Test Accuracy: 0.9599999785423279
```