```
import zipfile

try:
    with zipfile.ZipFile('/content/archive.zip', 'r') as zip_ref:
        print("The file is a valid ZIP archive.")
        # List files in the archive
        print("Files in the archive:", zip_ref.namelist())
except zipfile.BadZipFile:
    print("The file is not a valid ZIP archive or is corrupted.")
```

```
The file is a valid ZIP archive.
Files in the archive: ['dataset-resized/cardboard/cardboard1.jpg', 'dataset-resized/cardboard/cardboard10.jpg', 'dataset-resized/cardbo
```

```
import os
import shutil
!unzip -o "/content/archive.zip" -d /content/dataset
# Define paths
```

```
  inflating: /content/dataset/dataset-resized/trash/trash47.jpg
  inflating: /content/dataset/dataset-resized/trash/trash48.jpg
  inflating: /content/dataset/dataset-resized/trash/trash49.jpg
  inflating: /content/dataset/dataset-resized/trash/trash5.jpg
  inflating: /content/dataset/dataset-resized/trash/trash50.jpg
  inflating: /content/dataset/dataset-resized/trash/trash51.jpg
  inflating: /content/dataset/dataset-resized/trash/trash52.jpg
  inflating: /content/dataset/dataset-resized/trash/trash53.jpg
  inflating: /content/dataset/dataset-resized/trash/trash54.jpg
  inflating: /content/dataset/dataset-resized/trash/trash55.jpg
  inflating: /content/dataset/dataset-resized/trash/trash56.jpg
  inflating: /content/dataset/dataset-resized/trash/trash57.jpg
  inflating: /content/dataset/dataset-resized/trash/trash58.jpg
  inflating: /content/dataset/dataset-resized/trash/trash59.jpg
  inflating: /content/dataset/dataset-resized/trash/trash6.jpg
  inflating: /content/dataset/dataset-resized/trash/trash60.jpg
  inflating: /content/dataset/dataset-resized/trash/trash61.jpg
  inflating: /content/dataset/dataset-resized/trash/trash62.jpg
  inflating: /content/dataset/dataset-resized/trash/trash63.jpg
  inflating: /content/dataset/dataset-resized/trash/trash64.jpg
  inflating: /content/dataset/dataset-resized/trash/trash65.jpg
  inflating: /content/dataset/dataset-resized/trash/trash66.jpg
  inflating: /content/dataset/dataset-resized/trash/trash67.jpg
  inflating: /content/dataset/dataset-resized/trash/trash68.jpg
  inflating: /content/dataset/dataset-resized/trash/trash69.jpg
  inflating: /content/dataset/dataset-resized/trash/trash7.jpg
  inflating: /content/dataset/dataset-resized/trash/trash70.jpg
  inflating: /content/dataset/dataset-resized/trash/trash71.jpg
  inflating: /content/dataset/dataset-resized/trash/trash72.jpg
  inflating: /content/dataset/dataset-resized/trash/trash73.jpg
  inflating: /content/dataset/dataset-resized/trash/trash74.jpg
  inflating: /content/dataset/dataset-resized/trash/trash75.jpg
  inflating: /content/dataset/dataset-resized/trash/trash76.jpg
  inflating: /content/dataset/dataset-resized/trash/trash77.jpg
  inflating: /content/dataset/dataset-resized/trash/trash78.jpg
  inflating: /content/dataset/dataset-resized/trash/trash79.jpg
  inflating: /content/dataset/dataset-resized/trash/trash8.jpg
  inflating: /content/dataset/dataset-resized/trash/trash80.jpg
  inflating: /content/dataset/dataset-resized/trash/trash81.jpg
  inflating: /content/dataset/dataset-resized/trash/trash82.jpg
  inflating: /content/dataset/dataset-resized/trash/trash83.jpg
  inflating: /content/dataset/dataset-resized/trash/trash84.jpg
  inflating: /content/dataset/dataset-resized/trash/trash85.jpg
  inflating: /content/dataset/dataset-resized/trash/trash86.jpg
  inflating: /content/dataset/dataset-resized/trash/trash87.jpg
  inflating: /content/dataset/dataset-resized/trash/trash88.jpg
  inflating: /content/dataset/dataset-resized/trash/trash89.jpg
  inflating: /content/dataset/dataset-resized/trash/trash9.jpg
  inflating: /content/dataset/dataset-resized/trash/trash90.jpg
  inflating: /content/dataset/dataset-resized/trash/trash91.jpg
  inflating: /content/dataset/dataset-resized/trash/trash92.jpg
  inflating: /content/dataset/dataset-resized/trash/trash93.jpg
  inflating: /content/dataset/dataset-resized/trash/trash94.jpg
  inflating: /content/dataset/dataset-resized/trash/trash95.jpg
  inflating: /content/dataset/dataset-resized/trash/trash96.jpg
  inflating: /content/dataset/dataset-resized/trash/trash97.jpg
  inflating: /content/dataset/dataset-resized/trash/trash98.jpg
  inflating: /content/dataset/dataset-resized/trash/trash99.jpg
```

```
!pip install ultralytics
```

```
from ultralytics import YOLO
from sklearn.metrics import accuracy_score
import numpy as np
import os
import shutil
from sklearn.model_selection import train_test_split


dataset_path = '/content/dataset/dataset-resized'  # Path to your original dataset
output_path = '/content/yolo_dataset'  # Path to save the YOLO-formatted dataset

# Create YOLO dataset structure
os.makedirs(os.path.join(output_path, 'images', 'train'), exist_ok=True)
os.makedirs(os.path.join(output_path, 'images', 'val'), exist_ok=True)
os.makedirs(os.path.join(output_path, 'labels', 'train'), exist_ok=True)
os.makedirs(os.path.join(output_path, 'labels', 'val'), exist_ok=True)

# Class names and their corresponding IDs
class_names = ['cardboard', 'glass', 'metal', 'paper', 'plastic', 'trash']
class_ids = {name: idx for idx, name in enumerate(class_names)}

# Collect all images and their labels
images = []
labels = []

for class_name in class_names:
    class_folder = os.path.join(dataset_path, class_name)
    for image_name in os.listdir(class_folder):
        if image_name.endswith('.jpg') or image_name.endswith('.png'):
            images.append(os.path.join(class_folder, image_name))
            labels.append(class_name)

# Split into train and val sets (80% train, 20% val)
train_images, val_images, train_labels, val_labels = train_test_split(images, labels, test_size=0.2, random_state=42)

# Function to create YOLO label files
def create_yolo_label(image_path, label, output_label_path):
    # Get image dimensions (assuming all images are the same size)
    # You can use OpenCV to get the exact dimensions if needed
    image_width, image_height = 640, 640  # Replace with actual dimensions if known

    # Create YOLO label file
    with open(output_label_path, 'w') as f:
        class_id = class_ids[label]
        x_center, y_center, width, height = 0.5, 0.5, 1.0, 1.0  # Example: Placeholder values
        f.write(f"{class_id} {x_center} {y_center} {width} {height}\n")

# Copy images and create label files for train set
for image_path, label in zip(train_images, train_labels):
    image_name = os.path.basename(image_path)
    shutil.copy(image_path, os.path.join(output_path, 'images', 'train', image_name))
    label_path = os.path.join(output_path, 'labels', 'train', image_name.replace('.jpg', '.txt').replace('.png', '.txt'))
    create_yolo_label(image_path, label, label_path)

# Copy images and create label files for val set
for image_path, label in zip(val_images, val_labels):
    image_name = os.path.basename(image_path)
    shutil.copy(image_path, os.path.join(output_path, 'images', 'val', image_name))
    label_path = os.path.join(output_path, 'labels', 'val', image_name.replace('.jpg', '.txt').replace('.png', '.txt'))
    create_yolo_label(image_path, label, label_path)

print("Dataset conversion complete!")
```

```
    Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
  Attempting uninstall: nvidia-curand-cu12
    Found existing installation: nvidia-curand-cu12 10.3.6.82
    Uninstalling nvidia-curand-cu12-10.3.6.82:
      Successfully uninstalled nvidia-curand-cu12-10.3.6.82
  Attempting uninstall: nvidia-cufft-cu12
    Found existing installation: nvidia-cufft-cu12 11.2.3.61
    Uninstalling nvidia-cufft-cu12-11.2.3.61:
      Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
  Attempting uninstall: nvidia-cuda-runtime-cu12
    Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
    Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-nvrtc-cu12
    Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
    Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-cupti-cu12
    Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
    Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
  Attempting uninstall: nvidia-cublas-cu12
    Found existing installation: nvidia-cublas-cu12 12.5.3.2
    Uninstalling nvidia-cublas-cu12-12.5.3.2:
      Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
  Attempting uninstall: nvidia-cusparse-cu12
    Found existing installation: nvidia-cusparse-cu12 12.5.1.3
    Uninstalling nvidia-cusparse-cu12-12.5.1.3:
      Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
  Attempting uninstall: nvidia-cudnn-cu12
    Found existing installation: nvidia-cudnn-cu12 9.3.0.75
    Uninstalling nvidia-cudnn-cu12-9.3.0.75:
      Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
  Attempting uninstall: nvidia-cusolver-cu12
    Found existing installation: nvidia-cusolver-cu12 11.6.3.83
    Uninstalling nvidia-cusolver-cu12-11.6.3.83:
      Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
Successfully installed nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.127 nvidia-cuda-runtim
Creating new Ultralytics Settings v0.0.6 file ✅
View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralytics/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://docs.ultralytics.com/
Dataset conversion complete!
```

```
!ls /content/dataset
```

```
dataset-resized
```

```
import yaml

data_yaml = {
    'train': '/content/yolo_dataset/images/train',
    'val': '/content/yolo_dataset/images/val',
    'nc': 6,  # Number of classes
    'names': ['cardboard', 'glass', 'metal', 'plastic', 'paper', 'trash']  # Class names
}

with open('data.yaml', 'w') as outfile:
    yaml.dump(data_yaml, outfile, default_flow_style=False)


# Load a pre-trained YOLOv8 model
model = YOLO('yolov8n.pt')  # You can use 'yolov8s.pt', 'yolov8m.pt', etc.

# Train the model
results = model.train(data='data.yaml', epochs=150, imgsz=640, batch=16)
```

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
    146/150      4.86G    0.02649    0.06318     0.8566          5        640: 100%|██████████| 127/127 [00:31<00:00,  4.02it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 16/16 [00:03<00:00,  4.61i

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
    147/150      4.87G    0.02868    0.06614     0.8542          5        640: 100%|██████████| 127/127 [00:30<00:00,  4.12it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 16/16 [00:04<00:00,  3.57i

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
    148/150       4.9G    0.02528    0.05713      0.848          5        640: 100%|██████████| 127/127 [00:31<00:00,  3.97it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 16/16 [00:03<00:00,  4.40i

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
    149/150      4.91G    0.02327    0.05372     0.8498          5        640: 100%|██████████| 127/127 [00:31<00:00,  4.09it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 16/16 [00:03<00:00,  4.65i

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
    150/150      4.93G    0.02504    0.06115     0.8539          5        640: 100%|██████████| 127/127 [00:31<00:00,  4.04it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 16/16 [00:04<00:00,  3.93i

150 epochs completed in 1.539 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 6.2MB
Optimizer stripped from runs/detect/train/weights/best.pt, 6.2MB

Validating runs/detect/train/weights/best.pt...
Ultralytics 8.3.93 🚀 Python-3.11.11 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 72 layers, 3,006,818 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 16/16 [00:04<00:00,  3.40i
                   all        506        506       0.92      0.909      0.974      0.974
             cardboard         71         71      0.916      0.944      0.985      0.985
                 glass        107        107      0.988      0.832      0.977      0.977
                 metal         87         87      0.902      0.908      0.975      0.975
               plastic        118        118      0.982      0.906      0.992      0.992
                 paper         99         99      0.895      0.865       0.96       0.96
                 trash         24         24      0.837          1      0.954      0.954
Speed: 0.3ms preprocess, 1.9ms inference, 0.0ms loss, 2.0ms postprocess per image
Results saved to runs/detect/train
```

```
metrics = model.val()
```

```
Ultralytics 8.3.93 🚀 Python-3.11.11 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 72 layers, 3,006,818 parameters, 0 gradients, 8.1 GFLOPs
val: Scanning /content/yolo_dataset/labels/val.cache... 506 images, 0 backgrounds, 0 corrupt: 100%|██████████| 506/506 [00:00<?, ?it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 32/32 [00:05<00:00,  5.39it/
                   all        506        506      0.934      0.895      0.974      0.974
             cardboard         71         71      0.932      0.944      0.985      0.985
                 glass        107        107      0.995      0.785      0.977      0.977
                 metal         87         87      0.908      0.905      0.975      0.975
               plastic        118        118      0.982      0.901      0.992      0.992
                 paper         99         99      0.903      0.848      0.959      0.959
                 trash         24         24      0.888      0.987      0.954      0.954
Speed: 0.7ms preprocess, 3.5ms inference, 0.0ms loss, 1.3ms postprocess per image
Results saved to runs/detect/train2
```