LMCD-OR Dataset User's Guide

Welcome to the LMCD-OR dataset! The LMCD-OR contains 3,818 oral x-ray images from local hospital databases and is primarily used for image classification and disease diagnosis tasks. The dataset consists of original DICOM oral x-ray images from local healthcare facilities, carefully labeled to cover general categories and specific secondary disease diagnoses typical of primary dental care. This dataset is expected to provide more detailed, accurate and reliable imaging support information for the diagnosis, treatment outcome and prognosis of oral health conditions.

Guidelines for use

Step 1: Download Data

1.1 Visit the official LMCD-OR dataset website at http:/dentaldataset.zeroacademy.net/ or the Kaggle page.
1.2 Locate the dataset download link on the page and follow the prompts to download the dataset file (usually a compressed file, e.g., in .zip format).

Step 2: Unzip the data

2.1 Once the download is complete, extract the zip file to your local storage.
2.2 Upon decompression, you will see a folder containing 3,818 oral x-ray images. Each image file is in DICOM format, , and is accompanied by a corresponding labeled file.

Step 3: Data Preprocessing

3.1 Install the necessary Python libraries, such as pydicom, numpy, and opencv-python, for reading and processing DICOM images.
3.2 We provide easy-to-use sample data preprocessing code, please refer to use:

```
pip install pydicom numpy opencv-python
import numpy
import pydicom
import cv2
def dicom_to_image(dicom_path, output_path):
    ds = pydicom.dcmread(dicom_path)
    img = ds.pixel_array
    cv2.imwrite(output_path, img)
dicom_to_image('path/to/dicom/file', 'output/path/to/image.png')
```

Step 4: Establishing a baseline model

4.1 Prepare the dataset by dividing the images and labels into training and test sets.
Build a multiclassification convolutional neural network (CNN) model using a deep learning framework such as TensorFlow or PyTorch.
4.2 We have provided easy-to-use sample code for modeling the baseline, please refer to its use:

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense model = Sequential([ Conv2D(32, (3, 3), activation='relu', input_shape=(image_height, image_width, 1)), MaxPooling2D((2, 2)), Flatten(), Dense(128, activation='relu'), Dense(num_classes, activation='softmax') ])
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

Step 5: Training the model

5.1 Train the model using the prepared training data.
5.2 We provide sample code for training models that are easy to use, please refer to the use:

```
model.fit(train_images, train_labels, epochs=10, validation_data=(test_images, test_labels))
```

Step 6: Model Evaluation and Testing

6.1 After training is complete, the performance of the model is evaluated using a test set.
6.2 We provide easy-to-use sample data preprocessing code, please refer to use:

```
test_loss, test_acc = model.evaluate(test_images, test_labels)
print(f "Test accuracy: {test_acc}")
```

We hope this user guide will help you make the most of the LMCD-OR dataset and facilitate the smooth running of your research. If you have any questions or need further assistance, please feel free to contact us.