

```
import numpy as np
import pandas as pd

import os
```

```
import os
import torch
import pandas as pd
import numpy as np
from sklearn.utils import shuffle
from PIL import Image
import torchvision
from torchvision import datasets, transforms
```

```
        self.labels = {label: idx for idx, label in enumerate(labels)}
    def get_label(self, idx):
        return list(self.labels.keys())[idx]
    def get_idx(self, label):
        return self.labels.get(label)
```

Initializing label encoder with 21 classes and testing its functionality

```
encoder_21 = Label_encoder(classes_21)
```

```
encoder_21.get_label(0), encoder_21.get_idx( encoder_21.get_label(0) )
```

Printing each class with its corresponding index

```
for i in range(21):
```

```
    print(encoder_21.get_label(i), encoder_21.get_idx( encoder_21.get_label(i) ))
```

Defining a custom dataset class for handling image data

```
class Food21(Dataset):
```

```
    def __init__(self, dataframe, transform=None):
```

apple_pie 0
baby_back_ribs 1
baklava 2
beef_carpaccio 3
beef_tartare 4
beet_salad 5
beignets 6
bibimbap 7
bread_pudding 8
breakfast_burrito 9
bruschetta 10

```

def prep_df(path: str) -> pd.DataFrame:
    array = open(path, 'r').read().splitlines()
    # Getting the full path for the images
    img_path = "/kaggle/input/food-101/food-101/food-101/images/"
    full_path = [img_path + img + ".jpg" for img in array]
    # Splitting the image index from the label
    imgs = []
    for img in array:
        img = img.split('/')
        imgs.append(img)
    imgs = np.array(imgs)
    for idx, img in enumerate(imgs):
        if encoder_21.get_idx(img[0]) is None:
            imgs[idx, 0] = "other"
    # Converting the array to a data frame
    imgs = pd.DataFrame(imgs[:, 0], imgs[:, 1], columns=['label'])
    # Adding the full path to the data frame
    imgs['path'] = full_path

```

label	
apple_pie	750
baby_back_ribs	750
baklava	750
beef_carpaccio	750
beef_tartare	750
beet_salad	750
beignets	750
bibimbap	750
bread_pudding	750
breakfast_burrito	750
bruschetta	750
caesar_salad	750
cannoli	750
caprese_salad	750
carrot_cake	750
ceviche	750

```

plt.figure(figsize=(20, 5))
num_rows = 3
num_cols = 8
for idx in range(num_rows * num_cols):
    random_idx = np.random.randint(0, train_imgs.shape[0])
    img = plt.imread(train_imgs.path.iloc[random_idx])
    label = train_imgs.label.iloc[random_idx]
    ax = plt.subplot(num_rows, num_cols, idx + 1)
    plt.imshow(img)
    plt.title(label)
    plt.axis("off")

```

