

Практическое задание №1

Установка необходимых пакетов:

```
In [ ]: pip install -q tensorflow
pip install -q tqdm
pip install -q ipynb-pygments-highlighter
```

Мониторинг Вашего Google Drive в текущую секунду:

```
In [ ]: from google.colab import drive
drive.mount('/content/drive', force_remount=True)
```

В параметре Ю-PROJECT_DIR необходимо прописать путь к директории на Google Drive. Это пригодится при сохранении модели.

```
In [ ]: PROJECT_DIR = '/pracs_m'
```

Константы, которые пригодятся в коде далее:

```
In [ ]: EVALUATE_ONLY = True
TEST_ON_LARGE_DATASET = True
IMAGE_CLASSES = ('AIR', 'BACK', 'BEET', 'LIM', 'MUC', 'MS', 'NORM', 'STR', 'TUM')
```

Ссылки (drive идентификаторы) на предоставленные наборы данных:

```
In [ ]: DATASETS_LINKS = {
    'train_small': 'https://drive.google.com/drive/folders/1w6aF6C208',
    'train_tiny': 'https://drive.google.com/drive/folders/1w6aF6C208',
    'test': 'https://drive.google.com/drive/folders/1w6aF6C208',
    'test_small': 'https://drive.google.com/drive/folders/1w6aF6C208',
    'test_tiny': 'https://drive.google.com/drive/folders/1w6aF6C208'
}
```

Импорти необходимых зависимостей:

```
In [ ]: !pip install -q libtiff
```

```
In [ ]: from pathlib import Path
from listify import Listify
import numpy as np
import tensorflow as tf
from typing import List
from tqdm.notebook import tqdm
from time import sleep
from PIL import Image
import PythonDisplay
from sklearn.metrics import balanced_accuracy_score
from sklearn.model_selection import train_test_split
import glob
import matplotlib.pyplot as plt
import time
```

Класс Dataset

Предназначен для работы с наборами данных, хранящихся на Google Drive, обеспечивает чтение изображений и соответствующих меток, а также формирование пакетов (батчей).

```
In [ ]: class Dataset:
    def __init__(self, name):
        self.name = name
        self.is_loaded = False
        if not Path(f'{name}.npz').exists():
            url = f'https://drive.google.com/drive/folders/{DATASETS_LINKS[name]}'
            output = f'{name}.npz'
            gdown.download(url, output, quiet=False)
            print(f'Loading dataset {self.name} from npz:')
            np.npz = np.load(f'{name}.npz')
            self.images = np.array(self.data)
            self.labels = np.array(self.labels)
            self.n_files = self.images.shape[0]
            self.is_loaded = True
            print(f'Done. Dataset {name} consists of {self.n_files} images.')

    def image(self, i):
        # read i-th image in dataset and return it as numpy array
        if self.is_loaded:
            return self.images[i]
        else:
            return self.images[i]

    def images(self, n=None):
        # sequential access to images inside dataset (is needed for testing)
        for i in range(self.n_files if n is None else n):
            yield self.image(i)

    def random_image_with_label(self):
        # get random image with label from dataset
        i = np.random.randint(self.n_files)
        return self.image(i), self.labels[i]

    def random_batch_with_label(self, n):
        # create random batch of images with labels (is needed for training)
        indices = np.random.choice(self.n_files, n)
        imgs = []
        for i in indices:
            img = self.image(i)
            imgs.append(self.image(i))
            logits = np.array(self.labels[i] for i in indices)
        return np.stack(imgs), logits

    def image_with_label(self, i, info):
        # return i-th image with label from dataset
        return self.image(i), self.labels[i]
```

Загрузка датасетов

Загружаем датасеты:

- train_tiny = test_tiny для отладки построенной модели
- train_small = test_small для первичного тестирования модели
- train = test для финального тестирования модели

```
In [ ]: d_train_tiny = Dataset('train_tiny')
d_test_tiny = Dataset('test_tiny')
```

```
In [ ]: d_train_small = Dataset('train_small')
d_test_small = Dataset('test_small')
```

```
In [ ]: d_train = Dataset('train')
d_test = Dataset('test')
```

Класс Metrics

Реализует метрики точности, используемые для оценивания модели:

1. Точность.
2. Сбалансированную точность.

```
In [ ]: class Metrics:
    @staticmethod
    def accuracy(gt, List[int], pred, List[int]):
        assert len(gt) == len(pred), 'gt and prediction should be of equal length'
        return sum(int(i) == j for i, j in zip(gt, pred)) / len(gt)

    @staticmethod
    def accuracy_balanced(gt, List[int], pred, List[int]):
        return balanced_accuracy_score(gt, pred)

    @staticmethod
    def print_metrics_for_info(self, List[int], info: str):
        print(f'Metrics for {info}:')
        print(f'Accuracy: {self.accuracy(gt, pred)}')
        print(f'Balanced accuracy: {self.accuracy_balanced(gt, pred)}')
```

Построение модели классификации

В данном разделе представлены примеры работы аугментации данных, а также реализован сам класс Model, содержащий модель классификации.

Аугментация данных

```
In [ ]: x_val = aугментация_данные
data_augmentation_layer = tf.keras.Sequential([
    tf.keras.layers.RandomFlip('horizontal'),
    tf.keras.layers.RandomFlip('vertical'),
    tf.keras.layers.RandomRotation(0.2),
])
```

```
In [ ]: plt.figure(figsize=(8, 8))
image = d_train.random_image_with_label()
for i in range(9):
    ax = plt.subplot(3, 3, i + 1)
    augmented_image = data_augmentation_layer(tf.expand_dims(image, 0))
    plt.imshow(augmented_image[0])
    plt.axis('off')
```

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.

Изображения, полученные в результате аугментации данных.