Министерство Просвещения, Культуры и Исследований Республики Молдова

Технический университет Молдовы

Факультет Вычислительной Техники, Информатики и Микроэлектроники

Департамент Микроэлектроники и Биоинженерии

**Отчет**

**По предмету Programarea Declarativă**

**По лабораторной работе №3**

**Тема:** «Обработка данных.»

Выполнил Cuțar Alexandr (TI-218)

Проверил A.Leah

Кишинев 2023

# **Задания к самостоятельной работе**

**Вариант 3–10**

1. Написать скрипт для формирования текстового файла-аннотации собранного датасета. Файл-аннотация должен представлять собой csv-файл, в котором в первой колонке будет указан абсолютный путь к файлу, во второй колонке относительный путь относительно вашего Python-проекта, третья колонка будет содержать текстовое название класса (метку класса), к которому относится данный экземпляр.
2. Написать скрипт для копирования датасета в другую директорию таким образом, чтобы имена файлов содержали имя класса и его порядковый номер. То есть из dataset/class/0000.jpg должно получиться dataset/class\_0000.jpg. Для того чтобы осталась возможность определить принадлежность экземпляра к классу создать файл-аннотацию (как в пункте 1).
3. Написать скрипт, создающий копию датасета таким образом, чтобы каждый файл из сходного датасета получил случайный номер от 0 до 10000, и датасет представлял собой следующую структуру dataset/номер.jpg.
4. Написать скрипт, содержащий функцию, получающую на входе метку класса и возвращающую следующий экземпляр (путь к нему) этого класса. Экземпляры идут в любом порядке, но не повторяются. Когда экземпляры заканчиваются, функция возвращает None.

**Код main.py**

import os  
import shutil  
import random  
import csv  
  
  
def create\_annotation\_file(dataset\_dir, project\_dir):  
 with open('annotations.csv', 'w', newline='') as annotation\_file:  
 writer = csv.writer(annotation\_file)  
  
 for root, dirs, files in os.walk(dataset\_dir):  
 for filename in files:  
 class\_name = os.path.basename(root)  
 absolute\_path = os.path.join(root, filename)  
 relative\_path = os.path.relpath(absolute\_path, project\_dir)  
  
 writer.writerow([absolute\_path, relative\_path, class\_name])  
  
  
def copy\_dataset\_with\_new\_names(dataset\_dir, output\_dir):  
 os.makedirs(output\_dir, exist\_ok=True)  
  
 with open('annotations.csv', 'w', newline='') as annotation\_file:  
 writer = csv.writer(annotation\_file)  
  
 for root, dirs, files in os.walk(dataset\_dir):  
 for filename in files:  
 class\_name = os.path.basename(root)  
 old\_filepath = os.path.join(root, filename)  
 new\_filename = f'{class\_name}\_{filename}'  
 new\_filepath = os.path.join(output\_dir, new\_filename)  
  
 writer.writerow([new\_filepath, new\_filename, class\_name])  
 shutil.copy(old\_filepath, new\_filepath)  
  
  
def create\_dataset\_with\_random\_names(dataset\_dir, output\_dir):  
 random\_names\_dir = output\_dir+"\\random\_names"  
 os.makedirs(random\_names\_dir, exist\_ok=True)  
  
 random\_names = [str(random.randint(0, 10000)) for \_ in range(10000)]  
  
 for root, dirs, files in os.walk(dataset\_dir):  
 for filename in files:  
 class\_name = os.path.basename(root)  
 old\_filepath = os.path.join(root, filename)  
 random\_name = random\_names.pop()  
 new\_filename = f'{random\_name}.jpg'  
 new\_filepath = os.path.join(random\_names\_dir, new\_filename)  
 shutil.copy(old\_filepath, new\_filepath)  
  
  
def get\_next\_instance\_by\_class(class\_label, dataset\_dir):  
 class\_dir = os.path.join(dataset\_dir, class\_label)  
 if not os.path.exists(class\_dir):  
 return None  
  
 files = os.listdir(class\_dir)  
 random.shuffle(files)  
  
 while len(files) > 0:  
 next\_instance = os.path.join(class\_dir, files.pop())  
 print(f"Следующий экземпляр класса '{class\_label}': {next\_instance}")  
  
 print(f"Все экземпляры класса '{class\_label}' закончились")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 dataset\_dir = "D:\\Ucheba\\3 course\\PD\\lab2\\dataset"  
 project\_dir = "D:\\Ucheba\\3 course\\PD\\lab2"  
 output\_dir = "D:\\Ucheba\\3 course\\\PD\\lab3\\dataset"  
  
 create\_annotation\_file(dataset\_dir, project\_dir)  
 copy\_dataset\_with\_new\_names(dataset\_dir, output\_dir)  
 create\_dataset\_with\_random\_names(dataset\_dir, output\_dir)  
  
 class\_label = 'bay\_horse'  
 get\_next\_instance\_by\_class(class\_label, dataset\_dir)

**Результат выполнения**





