Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Санкт-Петербургский политехнический университет Петра Великого»
Институт компьютерных наук и кибербезопасности
Высшая школа технологий искусственного интеллекта

КУРСОВАЯ РАБОТА

УПРАВЛЕНИЕ МЕТАДАННЫМИ РЕЛЯЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ БАЗЫ ДАННЫХ POSTGRESQL

Выполнил:
студент гр. 5140201/50301 А.С. Тимофеев

Преподаватель:
доцент ВШТИИ ИКНК С.Г. Попов

Санкт-Петербург
2025

СОДЕРЖАНИЕ

ПОСТАНОВКА ЗАДАЧИ

В рамках работы необходимо реализовать приложение для распознавания возраста по фотографии используя CNN (convolutional neural network).

TO BE DONE...

1 СБОР ДАННЫХ

Для решения задачи распознавания возраста человека по фотографии лица был проведён анализ доступных открытых наборов данных (датасетов), содержащих изображения лиц с аннотированным возрастом.

1.1 Публичные датасеты

В рамках работы используются и рассматриваются следующие открытые датасеты:

– UTKFace;
– APPA-Real Age;

UTKFace

UTKFace - один из наиболее распространённых наборов данных для задач оценки возраста и пола. Он содержит более 20 000 изображений лиц людей в возрасте от 0 до 116 лет, снятых в различных условиях освещения, ракурсах и с разным фоном. [1] Каждое изображение имеет аннотацию, закодированную непосредственно в названии файла в формате:

`[age]_[gender]_[race]_[date_time].jpg`

– `age` — возраст человека (целое число);
– `gender` — пол (0 — мужчина, 1 — женщина);
– `race` — этническая принадлежность (0–4, пять категорий);
– `date_time` — метка времени.

Изображения имеют размер: 200 на 200 пикселей.

APPA-Real Age

APPA-Real Age - содержит 7591 изображение с указанием реального и предполагаемого возраста. Общее количество предполагаемых голосов составляет около 250 000. В среднем на каждое изображение приходится около

38 голосов, что делает средний предполагаемый возраст очень стабильным (0,3 стандартной ошибки от среднего значения). [2]

Каждое изображение имеет порядковый номер, а реальный возраст указан в отдельном `csv` файле:

– `image name` — порядковый номер изображения;
– `real_age` — реальный возраст;
– `apparent_age` — воспринимаемый возраст;

Изображения имеют различный размер.
В данной работе будем использовать только метки реального возраста.

Распределение возрастов людей на фотографиях приведено на Рисунке 1.

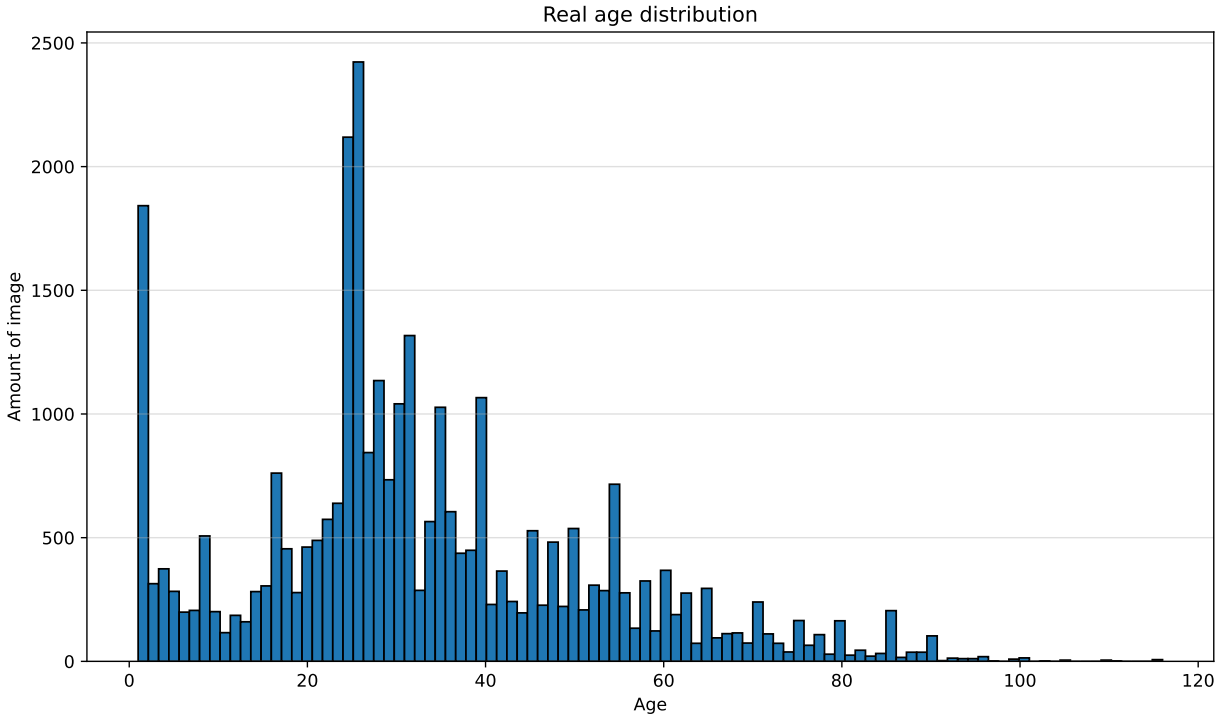

Рис. 1 — Распределения возраста лиц на фотографиях из датасетов APPA, UTKFace

2 ПРЕДОБРАБОТКА ДАННЫХ

Для выбранных датасетов будут выполнены следующие шаги для приведения данных к единому виду:

1. Загрузка данных из открытых источников.
2. Конкатенация датасетов с консистентным именем файла: `image_[number]_[real_age].jpg`
3. Создание `csv`-файла с метками для изображений.
4. Приведение изображений к единому размеру.
5. Разбиение на тестовую, валидационную и тестовые подвыборки.

Загрузка данных из открытых источников

Для загрузки датасетов был реализовать скрипт на языке `bash`. Исходный код скрипта приведен в Приложении А.

При успешном выполнении скрипта датасет из изображений представлен в предварительном формате:

```
...
datasets/
    UTKFace/        # Изображения метки( внутри названия файла)
    appa/
        *.csv       # .csv файлы с метками
        test/       # Изображения связаны с метко й через название
   файла
        train/
        val/
```

Листинг 2.1 — Иерархия файлов проекта

Конкатенация датасетов

Для конкатенации датасетов реализован скрипт на языке `Python` [3].

Скрипт собирает изображения из скачанных датасетов в единую директорию. Имена файлов имеют консистентные названия: `image_[number]_[real_age].jpg`.

Создание файла с метками

Для создания csv-файла с метками объектов реализован скрипт на языке Python [4].

В результате csv-файл имеет следующий вид:

```
,   image_name,             age
0,  image_23162_25.jpg,     25
1,  image_17662_78.jpg,     78
2,  image_06600_32.jpg,     32
```

Листинг 2.2 — csv-файл с метками объектов

Нормализация размеров изображений

Для нормализации размеров изображений реализован скрипт на языке Python [5]

Скрипт наивно (обычное сжатие или растягивание) изменяет размер изображения до заданного значения. В данной работе все изображения нормализованы к размеру 224 на 224 пикселя.

На Рисунке 2 приведен пример наивной нормализации изображения до заданного знчений.

Рис. 2 — Пример сжатия изображения из датасета APPA

Разделение датасета на подвыборки

Для разделения полученнго датасета на `test`, `train`, `valid`-подвыборки был реализован скрипт на языке `Python` [6]

Скрипт делит выборку целиком на K квантилей и равномерно забирает данные для подвыборок в соотношении:

– train: 70%;
– validation: 20%;
– test: 10%;

В результате сбора и предварительной обработки данных был получен датасет размером 31299. Каждое изображение приведено к единому размеру 224 на 224 пикселя. Создан `csv`-файл с метками для каждой фотографии лица.

ЗАКЛЮЧЕНИЕ

В результате курсовой работы разработано приложение для управления работы с метаданными реляционной системы управления базой данных PostgreSQL и реализован генератор подмножества `SELECT` запросов для заданных баз данных, а также хранение истории сгенерированных запросов.

Для решения данной задачи были выполнены следующие подзадачи:

1. Описание заданного окружения.
2. Создание хранилища метаданных:
   - проектирование схемы базы данных метаданных;
   - составление запросов к `information schema`;
   - реализация сохранения метаданных по DSN-строке подключения к СУБД;
3. Реализация графического интерфейса.
4. Разработка генератора подмножества `SELECT`.
5. Организация хранения истории запросов и возможности их повторного выполнения.

Следует отметить, что одним из недостатков рассматриваемой реализации является хранение истории запросов в базе данных метаданных, что частично нарушает разделения абстракций. Данный подход был выбран для упрощения реализации на этапе разработки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Zhang Z.*, *Song Y.*, *Qi H.* Age Progression/Regression by Conditional Adversarial Autoencoder // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2017. — P. 4352–4360.
2. *Agustsson E.*, *Timofte R.*, *Escalera S.*, [et al.]. Apparent and Real Age Estimation in Still Images with Deep Residual Regressors on the APPA-REAL Database // Proceedings of the 12th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG). — 2017. — P. 715–722.
3. *timofeevAS.* cnn-age-recognize. — 2025. — GitHub repository. https://github.com/timofeevAS/cnn-age-recognize/blob/main/src/dataset_preprocessing.py.
4. *timofeevAS.* cnn-age-recognize. — 2025. — GitHub repository. https://github.com/timofeevAS/cnn-age-recognize/blob/main/src/dataset_csv.py.
5. *timofeevAS.* cnn-age-recognize. — 2025. — GitHub repository. https://github.com/timofeevAS/cnn-age-recognize/blob/main/src/dataset_resize_image.py.
6. *timofeevAS.* cnn-age-recognize. — 2025. — GitHub repository. https://github.com/timofeevAS/cnn-age-recognize/blob/main/src/dataset_splitter.py.

ПРИЛОЖЕНИЕ А

```
#!/bin/bash
set -e

SCRIPT_DIR="$(cd␣"$(dirname "$0")"␣&&␣pwd)"
DATA_DIR="${SCRIPT_DIR}/datasets"
UTKFACE_DIR="${DATA_DIR}/UTKFace"
ZIP_PATH="${SCRIPT_DIR}/utkface-new.zip"
TMP_DIR="${DATA_DIR}/tmp"

echo "␣Installing␣UTKFace␣dataset..."

# Dont download archive if exists.
if [ ! -f "$ZIP_PATH" ]; then
echo "␣␣␣Downloading␣UTKFace..."
curl -L -o "$ZIP_PATH" "https://www.kaggle.com/api/v1/
   datasets/download/jangedoo/utkface-new"
fi

mkdir -p "$UTKFACE_DIR" "$TMP_DIR"
echo "␣␣Extracting␣only␣utkface_aligned_cropped/UTKFace..."
unzip -q -o "$ZIP_PATH" "utkface_aligned_cropped/UTKFace/*.
   jpg" -d "$TMP_DIR"

if [ -d "$TMP_DIR/utkface_aligned_cropped/UTKFace" ]; then
rsync -a "$TMP_DIR/utkface_aligned_cropped/UTKFace/" "
   $UTKFACE_DIR/"
echo "␣␣Moved␣$(ls␣"$UTKFACE_DIR"␣|␣wc␣-l)␣files␣to␣
   $UTKFACE_DIR"
else
echo "␣␣Expected␣folder␣not␣found␣inside␣archive"
exit 1
fi

echo "␣␣Delete␣temporary␣folder..."
rm -rf "$TMP_DIR"

echo "␣␣Installing␣UTKFaces␣succesful!"

```

```
    echo "␣Installing␣APPA␣dataset..."

    ZIP_PATH="${SCRIPT_DIR}/appa-real-release.zip"
    if [ ! -f "$ZIP_PATH" ]; then
    echo "␣␣Downloading␣APPA␣dataset..."
    curl -L -o "$ZIP_PATH" "https://data.chalearnlap.cvc.uab.cat/
       AppaRealAge/appa-real-release.zip"
    fi

    APPA_DIR="${DATA_DIR}/appa"

    echo "␣Extracting␣appa␣data␣set␣appa-real-release/..."
    unzip -q -o "$ZIP_PATH" \
    "appa-real-release/gt_avg_train.csv" \
    "appa-real-release/gt_test.csv" \
    "appa-real-release/gt_valid.csv" \
    "appa-real-release/train/*" \
    "appa-real-release/test/*" \
    "appa-real-release/valid/*" \
    -d $APPA_DIR

    if [ -d "$APPA_DIR/appa-real-release" ]; then
    rsync -a "$APPA_DIR/appa-real-release/" "$APPA_DIR"
    echo "␣Moved␣$(ls␣"$APPA_DIR/appa-real-release/"␣|␣wc␣-l)␣
       files␣to␣$APPA_DIR"
    else
    echo "␣Expected␣folder␣not␣found␣inside␣archive"
    exit 1
    fi

    echo "␣Delete␣temporary␣folder..."
    rm -rf "$APPA_DIR/appa-real-release"

    echo "␣Filtering␣appa␣dataset..."
    echo "␣Left␣only␣necessary␣pictures␣in␣storage...␣($APPA_DIR
       /train)"
    find "$APPA_DIR/train" -type f ! -name "*.jpg_face.jpg" -
       delete
    echo "␣Left␣only␣necessary␣pictures␣in␣storage...␣($APPA_DIR
       /test)"
    find "$APPA_DIR/test" -type f ! -name "*.jpg_face.jpg" -
       delete
```

```
    echo "␣Left␣only␣necessary␣pictures␣in␣storage...␣($APPA_DIR
       /valid)"
    find "$APPA_DIR/valid" -type f ! -name "*.jpg_face.jpg" -
       delete

    echo "␣Installing␣APPA␣real␣faces␣succesful!"
```

Листинг 2.3 — Bash-скрипт загрузки и распаковки датасета