

Проект: Определение возраста покупателей

Сетевой супермаркет «Хлеб-Соль» планирует внедрить систему компьютерного зрения для анализа фотографий покупателей, сделанных в прикассовой зоне. Новая система поможет решать следующие задачи:

- анализировать покупки и предлагать товары, соответствующие предпочтениям возрастной группы покупателей;
- контролировать соблюдение законодательства и добросовестность кассиров при продаже алкогольной продукции.

Для реализации этой системы необходимо разработать модель, способную определить возраст человека по фотографии. В нашем распоряжении имеется набор данных, включающий изображения людей с указанием их возраста.

Перед нами стоят следующие задачи:

1. Провести исследовательский анализ данных: изучить структуру и характеристики набора фотографий.
2. Подготовить данные для обучения нейронной сети.
3. Обучить модель и оценить ее точность на тестовых данных.

Эта система позволит супермаркету автоматизировать процесс анализа покупательского поведения и повысить уровень контроля при продаже возрастных товаров.

Исследовательский анализ данных

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from tensorflow.keras.layers import Conv2D, Flatten, GlobalAveragePooling2D, Dense, AvgPool2D
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.models import Sequential
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from PIL import Image
import os
```

Для начала загрузим наши данные

```
In [2]: labels = pd.read_csv('datasets/faces/labels.csv')
```

```
In [3]: display(labels.head())
print(labels.shape)
```

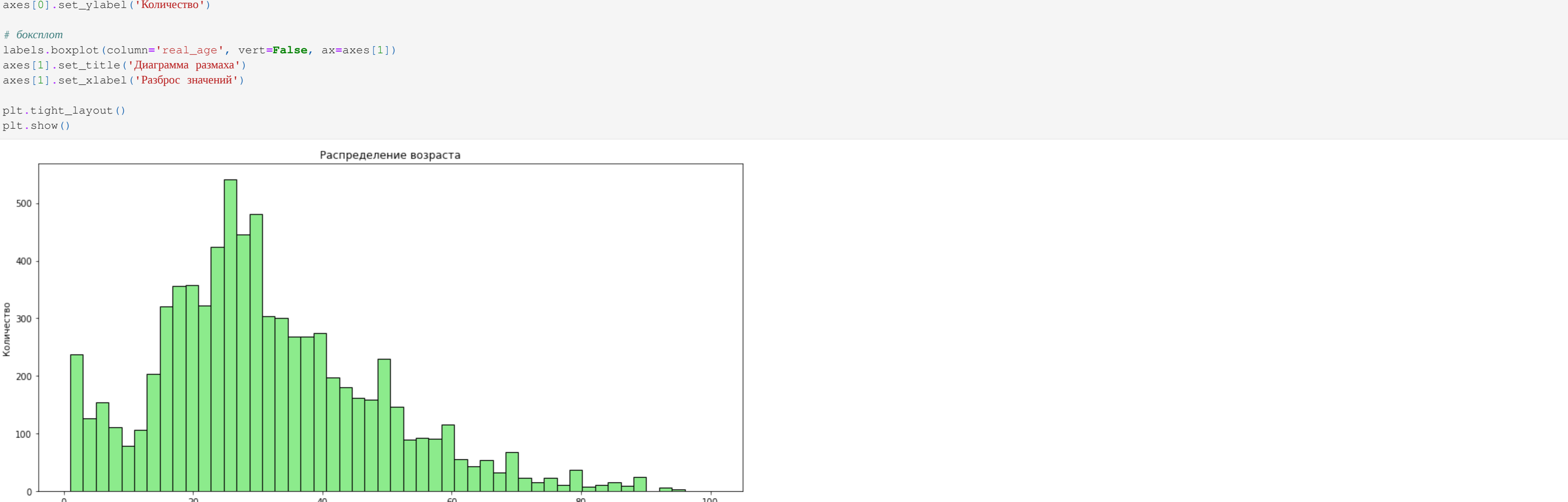
```
file_name  real_age
0  00000.jpg      4
1  00001.jpg     18
2  00002.jpg     80
3  00003.jpg     50
4  00004.jpg     17
(7591, 2)
```

Датасет состоит из 7591 записи, каждая из которых содержит имя файла (file_name) и связанный с ним возраст (real_age). Все данные полностью заполнены. Столбец file_name можно использовать для загрузки изображений, а real_age — как целевую переменную для предсказаний.

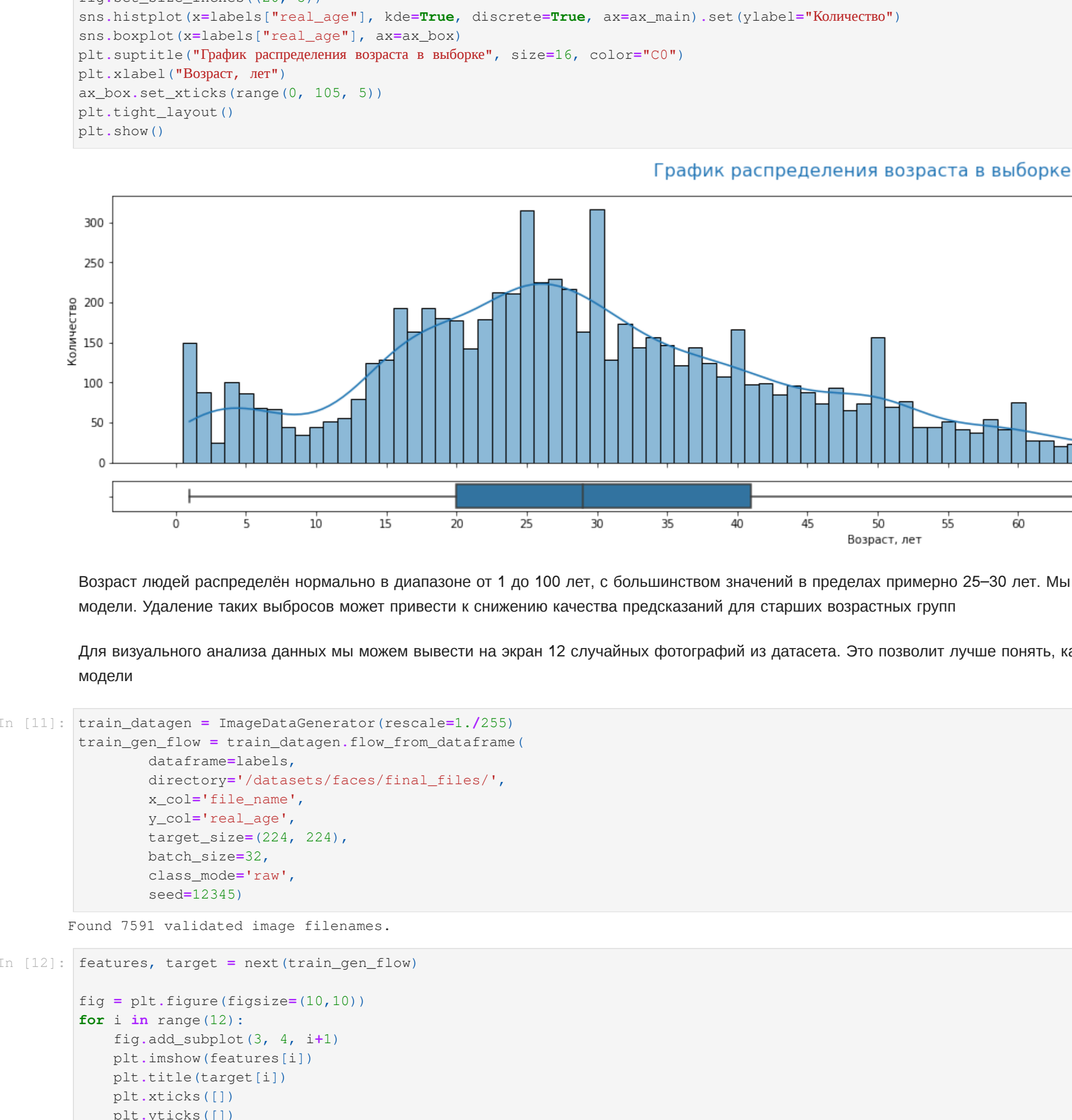
```
In [8]: labels.describe().T

Out[8]:
```

	count	mean	std	min	25%	50%	75%	max
real_age	7591.0	31.201159	17.14508	1.0	20.0	29.0	41.0	100.0



Построим такой же график распределения возраста, но более наглядный



Возраст людей распределен нормально в диапазоне от 1 до 100 лет, с наибольшим значением в пределах примерно 25–30 лет. Мы не будем удалять данные людей старше 80 лет, так как эти данные, несмотря на их редкость, могут быть важны для обучения модели. Удаление таких выбросов может привести к снижению качества предсказаний для старших возрастных групп.

Для визуального анализа данных мы можем вывести на экран 12 случайных фотографий из датасета. Это позволит лучше понять, как выглядят изображения, их качество, освещенность и присутствуют ли какие-либо проблемы, которые могут повлиять на обучение модели.



Посмотрим на формат изображений

```
In [13]: file_names = train_gen_flow.filenames[:5]

# Путь к директории с изображениями
image_dir = train_gen_flow.directory

# Выход формата изображений
for file_name in file_names:
    image_path = os.path.join(image_dir, file_name)
    with Image.open(image_path) as img:
        print(f'Файл: {file_name}, Формат: {img.format}')
```

Файл: 00000.jpg, Формат: JPEG
Файл: 00001.jpg, Формат: JPEG
Файл: 00002.jpg, Формат: JPEG
Файл: 00003.jpg, Формат: JPEG
Файл: 00004.jpg, Формат: JPEG

Итак, по данным приведенным выше мы можем сделать небольшие выводы:

Датасет содержит более 7.5 тыс. фотографий людей разного возраста и пола. Изображения имеют формат jpg, все фото квадратные, размер 224x224 пикселя. Большинство людей находятся в диапазоне возраста 25–30 лет. Пропуски и дубликаты в данных отсутствуют.

Фотографии представлены как в цвете, так и в черно-белом варианте. Лица расположены по центру кадра и находятся в вертикальном положении. На изображениях встречаются различные ракурсы, что означает, что лица могут быть сняты под разными углами, от более прямых до слегка наклонных. Также наблюдаются разные уровни освещенности: некоторые фотографии ярко освещены, в то время как на других присутствуют темные участки из-за недостаточного освещения. На некоторых изображениях могут быть тени, что может затруднить точное определение черт лица. Также встречаются засветы, когда части изображения из-за сильного света становятся слишком яркими и теряют детали. Также на некоторых фотографиях люди носят очки или головные уборы.

Датасет обладает достаточным объемом данных для обучения нейронной сети, такой как ResNet и не требует дополнительной обработки изображений. Однако для повышения качества предсказаний можно использовать методы аугментации для обучающих данных. Это позволит модели познакомиться с более разнообразными примерами и улучшить её обобщающую способность.

Обучение модели

Далее приступим к обучению модели.

```
def load_train(path):
    df = pd.read_csv(path + 'labels.csv')
    datagen = ImageDataGenerator(
        rescale=1./255,
        validation_split=0.25,
        horizontal_flip=True,
        rotation_range=45,
        width_shift_range=0.4,
        height_shift_range=0.4,
        zoom_range=0.2
    )

    train_data = datagen.flow_from_dataframe(
        dataframe=df,
        directory=path+'final_files',
        x_col='file_name',
        y_col='real_age',
        target_size=(224, 224),
        batch_size=20,
        seed=12345,
        subset='validation',
        class_mode='raw'
    )

    return train_data

def load_test(path):
    df = pd.read_csv(path + 'labels.csv')
    datagen = ImageDataGenerator(
        rescale=1./255,
        validation_split=0.25
    )

    test_data = datagen.flow_from_dataframe(
        dataframe=df,
        directory=path+'final_files',
        x_col='file_name',
        y_col='real_age',
        target_size=(224, 224),
        batch_size=20,
        seed=12345,
        subset='validation',
        class_mode='raw'
    )

    return test_data

def create_model(input_shape):
    backbone = ResNet50(input_shape=input_shape,
                        weights='datasets/keras_models/resnet50_weights_tf_dim_ordering_tf_anchors_notop.h5',
                        include_top=False)

    backbone.trainable = True

    model = Sequential()
    model.add(backbone)
    model.add(GlobalAveragePooling2D())
    model.add(Dense(1, activation='relu'))

    model.compile(optimizer=Adam(learning_rate=0.0001),
                  loss='mse',
                  metrics=['mae'])

    return model

def train_model(model, train_data, test_data, batch_size=12, epochs=12,
                steps_per_epoch=None, validation_steps=None):
    model.fit(train_data,
              validation_data=test_data,
              batch_size=batch_size, epochs=epochs,
              steps_per_epoch=len(train_data),
              validation_steps=len(test_data),
              verbose=2)

    return model

Found 5694 validated image filenames.
Found 1897 validated image filenames.
Train for 285 steps, validate for 95 steps

Epoch 1/12
285/285 - 136s - loss: 310.6598 - mae: 12.8509 - val_loss: 764.4649 - val_mae: 22.7309
Epoch 2/12
285/285 - 123s - loss: 145.3507 - mae: 9.1300 - val_loss: 238.6181 - val_mae: 11.1533
Epoch 3/12
285/285 - 126s - loss: 123.2071 - mae: 8.4320 - val_loss: 120.9947 - val_mae: 8.4798
Epoch 4/12
285/285 - 125s - loss: 113.9466 - mae: 8.1302 - val_loss: 167.0357 - val_mae: 9.3112
Epoch 5/12
285/285 - 126s - loss: 107.8987 - mae: 7.8530 - val_loss: 95.9549 - val_mae: 7.4100
Epoch 6/12
285/285 - 126s - loss: 96.9240 - mae: 7.4624 - val_loss: 124.0975 - val_mae: 8.5425
Epoch 7/12
285/285 - 120s - loss: 92.8201 - mae: 7.3517 - val_loss: 102.2473 - val_mae: 7.6898
Epoch 8/12
285/285 - 111s - loss: 85.0350 - mae: 6.9783 - val_loss: 87.1411 - val_mae: 6.9602
Epoch 9/12
285/285 - 112s - loss: 84.5725 - mae: 7.0246 - val_loss: 104.1859 - val_mae: 7.9031
Epoch 10/12
285/285 - 111s - loss: 81.6371 - mae: 6.8051 - val_loss: 105.3714 - val_mae: 7.6579
Epoch 11/12
285/285 - 111s - loss: 74.9100 - mae: 6.5989 - val_loss: 86.9609 - val_mae: 6.8726
Epoch 12/12
285/285 - 117s - loss: 69.4043 - mae: 6.3235 - val_loss: 80.0486 - val_mae: 6.6959

95/95 - 13s - loss: 80.0486 - mae: 6.6959
```

На этом этапе работы мы построили и обучили сверточную нейронную сеть на предоставленном датасете фотографий людей, чтобы предсказывать возраст по изображениям. Для подготовки тренировочной и тестовой выборки были использованы функции load_train(path) и load_test(path), включающие аугментацию данных: горизонтальное отражение, повороты, сдвиги и масштабирование. Эти техники помогли увеличить разнообразие данных и повысить устойчивость модели к различным трансформациям.

Для построения модели была создана функция create_model(input_shape), которая использует предварительно обученную ResNet50 в качестве основы и алгоритм обучения нейронной сети Adam. Мы добавили слой глобального усреднения (GlobalAveragePooling2D) и один полносвязный слой для предсказания возраста. В качестве функции потерь выбрали Mean Squared Error (MSE), поскольку задача предсказания возраста является задачей регрессии, а в качестве метрики использовали Mean Absolute Error (MAE).

Обучение модели осуществлялось с помощью функции train_model с заданными параметрами: размер batch, количество эпох, количество шагов на эпоху и шагов для валидации. Настроенная модель и предварительно обученная ResNet позволили добиться высоких результатов на тестовой выборке.

Итогом выполнения данного этапа стала обученная модель с высокой точностью, которая эффективно предсказывает приблизительный возраст на основе фотографий.

Анализ обученной модели

На основе полученных результатов можно сделать несколько выводов:

Для обучения использовались 5694 изображения, а для валидации — 1897 изображений. Задача — достичь значения MAE на валидационной выборке ниже 8 — была успешно выполнена. К концу обучения MAE составила 6.69, что означает, что погрешность предсказания возраста составляет ± 6.7 лет. Модель достигла значения MAE не больше 8 на валидационной выборке уже к 7-й эпохе, а к 12-й эпохе показатель MAE составил 6.69. Среднее время на одну эпоху составило около 2 минут.

Разница между значениями MAE на тренировочной выборке (6.32) и валидационной выборке (6.69) в конце обучения минимальна, что свидетельствует о хорошем обобщении модели. Показатели MSE (среднеквадратичной ошибки), используемой как функции потерь, также снижались, что подтверждает эффективность обучения.

Тем не менее, хотя мы достигли хороших результатов и успешно выполнили поставленную задачу, модель не достигла необходимой точности для использования в реальных бизнес-условиях. Это связано с тем, что датасет в основном состоит из фотографий людей среднего возраста, в нем недостаточно изображений детей, особенно школьников, и пожилых людей. Дополнение данных для этих возрастных групп улучшит точность модели.

Таким образом, модель показала хорошие результаты и может эффективно использоваться для анализа покупок и разработки персонализированных предложений в зависимости от возраста. Однако для задач, требующих высокой точности (например, продажа алкоголя), её точности недостаточно, но для анализа покупок и персонализированных предложений она подходит.

Вывод

Мы провели разработку и обучение модели для анализа фотографий покупателей в сетевом супермаркете «Хлеб-Соль» с целью решения двух задач: анализа покупок и предложения товаров на основе возраста покупателей, а также контроля соблюдения законодательства при продаже алкогольной продукции.

Сначала мы провели исследовательский анализ данных и выявили, что датасет, включает более 7.5 тыс. изображений людей разных возрастов и полов. Основная возрастная группа на изображениях — люди в возрасте 25–30 лет. Пропуски и дубликаты данных отсутствовали. Фотографии разнообразны по качеству и условиям съемки, что потребовало применения методов аугментации для улучшения качества данных для обучения модели.

После произвели обучение модели, которая основана на предварительно обученной ResNet50 с добавлением слоя глобального усреднения и полносвязного слоя для предсказания возраста. Для оптимизации использовался алгоритм Adam, а метрика — MAE.

Результат MAE на валидационной выборке составил 6.69 (погрешность ± 6.7 лет). Разница между MAE на тренировочной и валидационной выборках минимальна, что подтверждает хорошую обобщающую способность.

Однако модель не достигает необходимой точности для использования в реальных условиях, поскольку датасет не включает достаточно изображений детей и пожилых людей. Для задач, требующих высокой точности (например, продажа алкоголя), её точности недостаточно, но для анализа покупок и персонализированных предложений она подходит.