УДК 004

Импортирование, проверка готовой модели нейронной сети, обучение новой нейронной сети на основе готового датасета

Звайгзне Алексей Юрьевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье рассматривается процесс импортирование нейронной сети из системы Roboflow, проверка работы в Google colab, создание новой модели нейронной сети на основе готового датасета с использованием модели нейронной сети yolov5, проверка полученной нейронной сети.
Ключевые слова: Roboflow, yolov5, нейронная сеть, датасет, кузов, автомобиль, python, google colab

Importing, checking a ready-made neural network model, training a new neural network based on a ready-made dataset

Zvaigzne Alexey Yurievich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article discusses the process of importing a neural network from the Roboflow system, checking the work in Google colab, creating a new neural network model based on a ready-made dataset using the yolov5 neural network model, checking the resulting neural network.
Keywords: Roboflow, yolov5, neural network, dataset, body, car, python, google colab

1 Введение

1.1 Актуальность

Спрос на автомобили и связанные с ними услуги продолжает расти. Имеющиеся и новые игроки на рынке нуждаются в эффективных инструментах для обработки и классификации информации об автомобилях. Разработка модели, которая может определять тип кузова автомобиля по фотографии, может быть востребована в автомобильной промышленности, автомобильной торговле, страховании автомобилей и других смежных отраслях.

Развитие искусственного интеллекта и машинного обучения позволяет создавать более точные и эффективные модели для обработки и анализа изображений. Разработка классификационной модели на основе нейронных сетей и использование инструментов, таких как Roboflow, открывает новые

возможности для автоматизации процесса определения типа кузова автомобиля. Точная классификация типов кузовов автомобилей является важной задачей в различных сферах, таких как оценка стоимости автомобилей, инвентаризация автопарков, реклама и маркетинг автомобилей и другие. Разработка модели, которая способна точно определять тип кузова по фотографии, может значительно улучшить качество и эффективность работы в этих областях. Технологии компьютерного зрения, включая распознавание образов и классификацию изображений, продолжают развиваться. Множество исследований и разработок в области нейронных сетей и глубокого обучения позволяют создавать более точные и надежные модели для анализа изображений, в том числе для классификации типов кузовов автомобилей.

1.2Обзор исследований

А. Ю. Звайгзне в своей статье рассмотрел процесс создания модели классификационной нейронной сети с использованием изображений из готового датапака, процесс сбора нового датапака, обработка изображений в системе Roboflow [1]. Кью Лин в своей статье предложили облачную систему управления рабочими процессами, организующую конвейеры разработки роботов, улучшенных искусственным интеллектом [2]. Э. Бисогон в своей работе описал процесс создания моделей машинного обучения и глубокого обучения используя платформу Google colab [3]. Э. Каллиамваку в своей работе рассказал о особенностях платформы, множество проектов являются заброшенными и не реализованными, также то, что многие пользователи используют Github как файлообменник [4]. В. Вентонг объясняет принцип работы полностью сверточной модели нейронной сети yolov5 [5].

1.3Цель исследования

Импортировать и проверить работу готовой модели нейронной сети импортировать и создать новую модель нейронной сети на основе готового датасета

2 Материалы и методы

Для работы с платформой Roboflow, используется бесплатная среда разработки Google colab, работающая на языке программирования python, процесс взаимодействия происходит через API Roboflow, для создания модели новой нейронной сети используется готовый репозиторий с Github с встроенной сверточной моделью нейронной сети yolov5.

3 Результаты и обсуждения

Для импорта модели из roboflow в Google colab, на новом листе в ячейке кода указывается команда на установку библиотеки Roboflow (рис. 1).

```
%pip install Roboflow

Requirement already satisfied: Roboflow in /usr/local/lib/python3.10/dist-packages (1.1.0)
Requirement already satisfied: certifi==2022.12.7 in /usr/local/lib/python3.10/dist-packages (from Roboflow) (2022.12.7)
Requirement already satisfied: chardet==4.0.0 in /usr/local/lib/python3.10/dist-packages (from Roboflow) (4.0.0)
Requirement already satisfied: cycler==0.10.0 in /usr/local/lib/python3.10/dist-packages (from Roboflow) (0.10.0)
Requirement already satisfied: idna==2.10 in /usr/local/lib/python3.10/dist-packages (from Roboflow) (2.10)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.10/dist-packages (from Roboflow) (1.4.4)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from Roboflow) (3.7.1)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from Roboflow) (1.22.4)
Requirement already satisfied: opencv-python>=4.1.2 in /usr/local/lib/python3.10/dist-packages (from Roboflow) (4.7.0.72)
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from Roboflow) (8.4.0)
Requirement already satisfied: pyparsing==2.4.7 in /usr/local/lib/python3.10/dist-packages (from Roboflow) (2.4.7)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from Roboflow) (2.8.2)
Requirement already satisfied: python-dotenv in /usr/local/lib/python3.10/dist-packages (from Roboflow) (1.0.0)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from Roboflow) (2.27.1)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from Roboflow) (1.16.0)
Requirement already satisfied: supervision in /usr/local/lib/python3.10/dist-packages (from Roboflow) (0.11.1)
Requirement already satisfied: urllib3>=1.26.6 in /usr/local/lib/python3.10/dist-packages (from Roboflow) (1.26.16)
Requirement already satisfied: wget in /usr/local/lib/python3.10/dist-packages (from Roboflow) (3.2)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-packages (from Roboflow) (4.65.0)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from Roboflow) (6.0)
Requirement already satisfied: requests-toolbelt in /usr/local/lib/python3.10/dist-packages (from Roboflow) (1.0.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->Roboflow) (1.1.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->Roboflow) (4.40.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->Roboflow) (23.1)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests->Roboflow) (2.0.12)
```

Рисунок 1. Установка библиотеки Roboflow

Импортируется сама библиотека (рис.2), создается новая переменная rf которой присваивается уникальный ключ API для взаимодействия с рабочим пространством и проектами, в частности.

Переменная project определяет созданный проект для дальнейшего взаимодействия в системе.

Переменная model отвечает за выбор рабочей модели системы.

```
from roboflow import Roboflow

rf = Roboflow(api_key="x2q0R3cLof0cmo9bFaxV")
project = rf.workspace("type-auto").project("type-a-back-cars")
model = project.version(1).model

loading Roboflow workspace...
loading Roboflow project...
```

Рисунок 2. Импортирование библиотеки и подключение рабочей модели

Для проверки работы модели подобрано случайное изображение из интернета (рис. 3-4).

Рисунок 3. Случайная фотография автомобиля с типом кузова купе

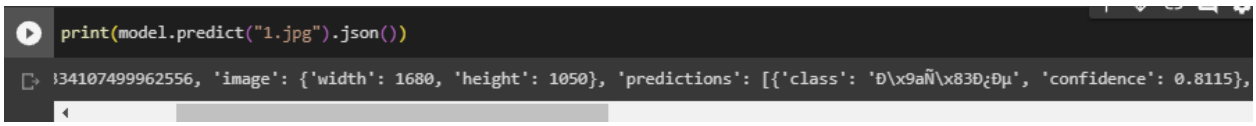

Рисунок 4. Результат работы модели нейросети

Нейросеть ошибочно предположила, что на данном изображении седан.

Для работы с dataset'ом используется модель yolov5, которая скачивается с Github, указывается директория и производится импортирование необходимых для работы библиотек (рис. 5).

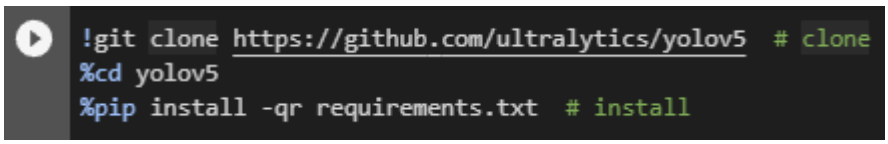

Рисунок 5. Установка yolov5

Для создания собственной модели нейросети на основе уже имеющегося датасета, через API происходит обращение к серверу, с указанием версии датасета (рис. 6).

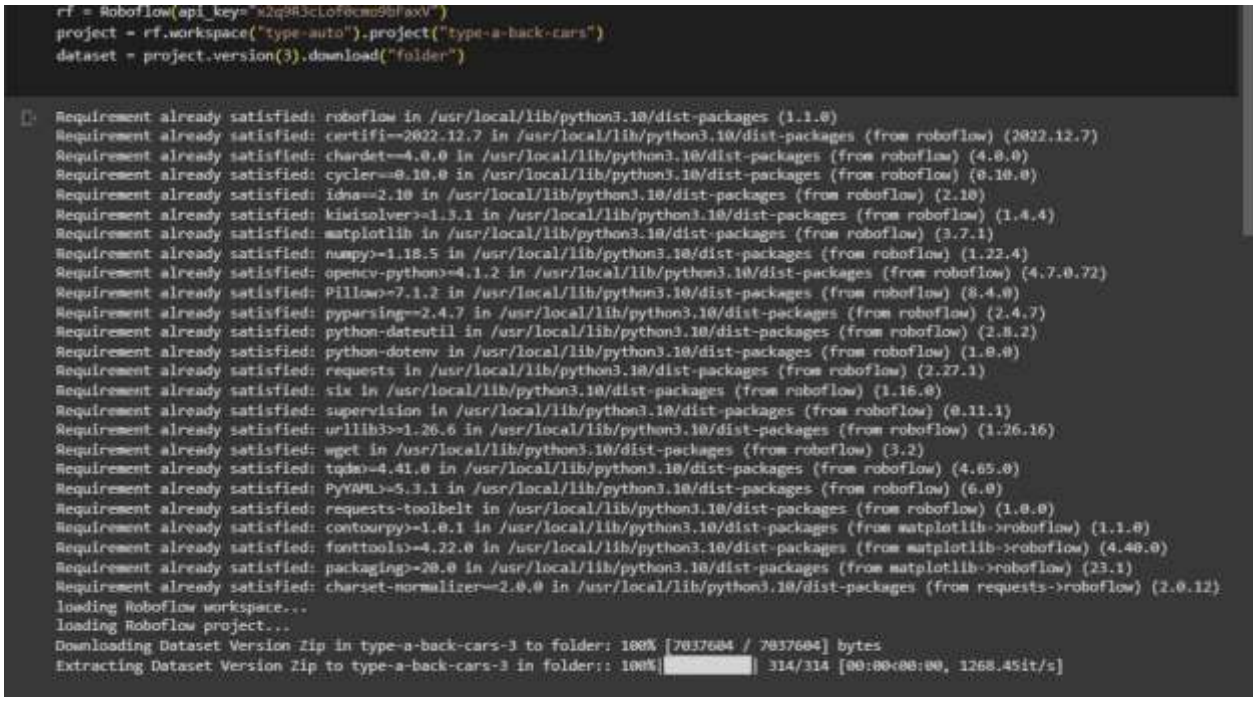

Рисунок 6. Скачивание датасета

Импорт библиотеки OS для работы с каталогами системы, создание нового каталога и перенос скачанного ранее датасета, выбор созданного каталога как рабочей директории (рис. 7).

```
import os
os.makedirs("../datasets/", exist_ok=True)
%cd ../datasets/

/content/datasets
```

Рисунок 7. Работа с каталогами

Переопределение датасета в переменную и присвоение каталогу датасета рабочего названия для дальнейшего обращения при работе через файловую систему (рис. 8).

```
[4] dataset_name = dataset.location.split(os.sep)[-1]
    os.environ["DATASET_NAME"] = dataset_name
```

Рисунок 8. Назначение рабочего каталога датасетом

Смена рабочей директории на yolov5 выполнение процесса обучения, для оптимальных значений установлено 200 эпох, размерность изображений 104 на 104 пикселя (рис. 9).

```
%cd ../yolov5
!python classify/train.py --model yolov5s-cls.pt --data $DATASET_NAME --epochs 200 --img 104 --pretrained weights/yolov5s-cls.pt

/content/yolov5
classify/train: model=yolov5s-cls.pt, data=type-a-back-cars-3, epochs=200, batch_size=64, imgsz=104, nosave=False, cache=None, device=
github: up to date with https://github.com/ultralytics/yolov5 ✅
YOLOv5 🚀 v7.0-189-ga453a45 Python-3.10.12 torch-2.0.1+cu118 CPU

TensorBoard: Start with 'tensorboard --logdir runs/train-cls', view at http://localhost:6006/
albumentations: RandomResizedCrop(p=1.0, height=104, width=104, scale=(0.08, 1.0), ratio=(0.75, 1.3333333333333333), interpolation=1),
Downloading https://github.com/ultralytics/yolov5/releases/download/v7.0/yolov5s-cls.pt to yolov5s-cls.pt...
100% 10.5M/10.5M [00:00<00:00, 93.5MB/s]

Model summary: 149 layers, 4176323 parameters, 4176323 gradients, 10.5 GFLOPs
optimizer: Adam(lr=0.001) with parameter groups 32 weight(decay=0.0), 33 weight(decay=5e-05), 33 bias
Image sizes 104 train, 104 test
Using 1 dataloader workers
Logging results to runs/train-cls/exp
Starting yolov5s-cls.pt training on type-a-back-cars-3 dataset with 3 classes for 200 epochs...

     Epoch   GPU_mem  train_loss   test_loss    top1_acc    top5_acc
     1/200        0G        1.32         1.1       0.333           1: 100% 4/4 [00:08<00:00,  2.02s/it]
     2/200        0G        1.09         1.1       0.333           1: 100% 4/4 [00:06<00:00,  1.57s/it]
     3/200        0G        1.08         1.1       0.333           1: 100% 4/4 [00:10<00:00,  2.75s/it]
     4/200        0G        1.08        1.11       0.333           1: 100% 4/4 [00:06<00:00,  1.73s/it]
     5/200        0G        1.11         1.1       0.333           1: 100% 4/4 [00:06<00:00,  1.71s/it]
     6/200        0G        1.09        1.11       0.333           1: 100% 4/4 [00:07<00:00,  1.84s/it]
     7/200        0G        1.05        1.11       0.333           1: 100% 4/4 [00:06<00:00,  1.72s/it]
     8/200        0G        1.06        1.11       0.333           1: 100% 4/4 [00:07<00:00,  1.93s/it]
     9/200        0G        1.13        1.11       0.333           1: 100% 4/4 [00:06<00:00,  1.55s/it]
    10/200        0G        1.12        1.11       0.333           1: 100% 4/4 [00:07<00:00,  1.95s/it]
    11/200        0G        1.07        1.11       0.333           1: 100% 4/4 [00:06<00:00,  1.55s/it]
    12/200        0G        1.06        1.11       0.333           1: 100% 4/4 [00:07<00:00,  1.92s/it]
    13/200        0G        1.05        1.13       0.333           1: 100% 4/4 [00:06<00:00,  1.56s/it]
    14/200        0G        1.04        1.13       0.333           1: 100% 4/4 [00:08<00:00,  2.02s/it]
    15/200        0G        1.08        1.12       0.333           1: 100% 4/4 [00:06<00:00,  1.57s/it]
    16/200        0G        1.03        1.11       0.333           1: 100% 4/4 [00:07<00:00,  1.94s/it]
    17/200        0G        1.06        1.11       0.333           1: 100% 4/4 [00:06<00:00,  1.57s/it]
    18/200        0G        1.09         1.1       0.333           1: 100% 4/4 [00:07<00:00,  1.93s/it]
    19/200        0G        1.08        1.09         0.4           1: 100% 4/4 [00:06<00:00,  1.56s/it]
    20/200        0G        1.05        1.08       0.433           1: 100% 4/4 [00:07<00:00,  1.93s/it]
    21/200        0G        1.05        1.05       0.433           1: 100% 4/4 [00:06<00:00,  1.55s/it]
    22/200        0G        1.04        1.06       0.433           1: 100% 4/4 [00:07<00:00,  1.91s/it]
    23/200        0G        1.05        1.06       0.433           1: 100% 4/4 [00:06<00:00,  1.55s/it]
```

Рисунок 9. Обучение модели нейронной сети

После завершения процесса обучения выводится кратка сводка указанием процентов точности (рис. 10).

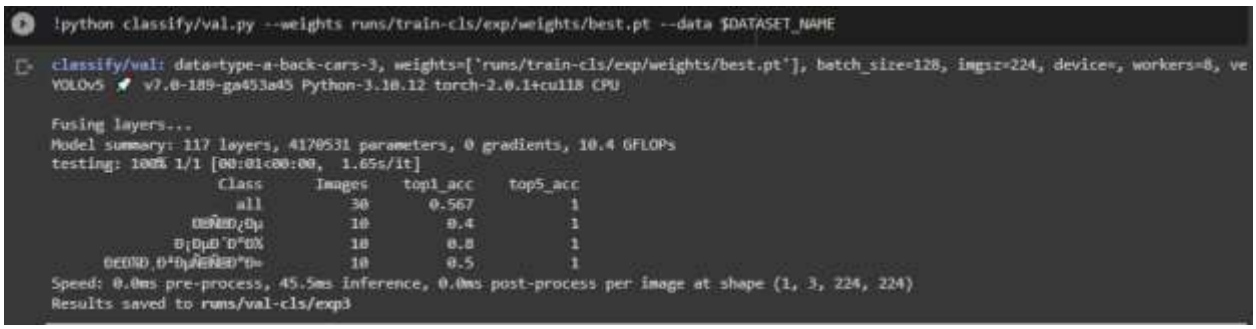

Рисунок 10. Кратка сводка по обученной модели

Проверка полученных результатов путем выбора изображения из массива тестовых картинок (рис. 11-12).

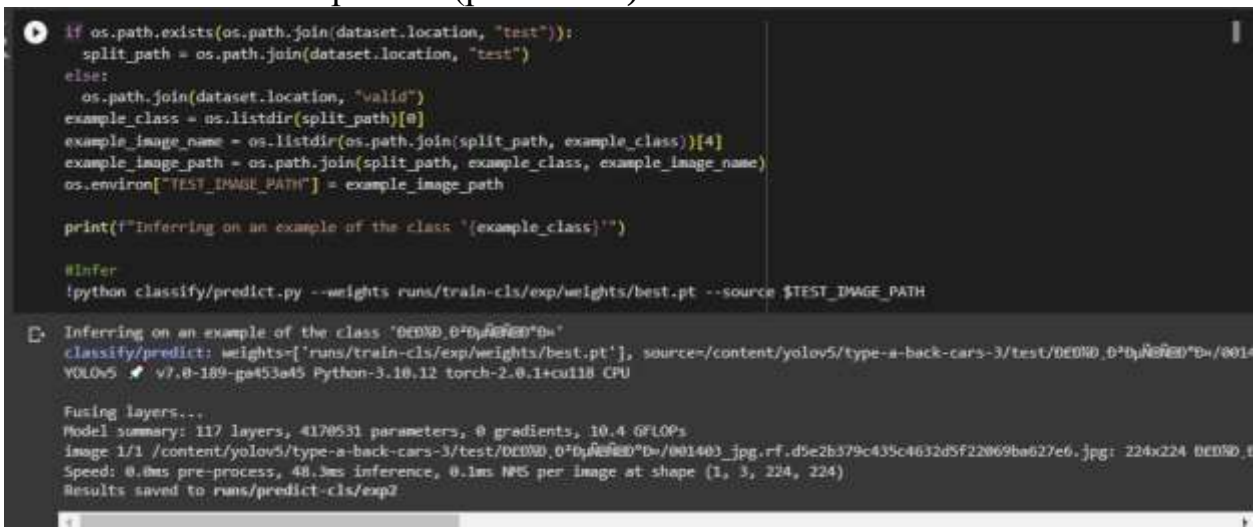

Рисунок 11. Код и результаты тестов

Рисунок 12. Графическое отображение полученных результатов

Выводы

Разработка классификационной модели для определения типа кузова автомобиля по фотографии является актуальной задачей, которая может быть решена с помощью нейронных сетей.

Использование предварительно обученных моделей, импортированных из Roboflow, может значительно ускорить процесс разработки и обучения модели, а также повысить ее точность. Roboflow предоставляет мощные инструменты для управления датасетами, предобработки данных, обучения моделей и экспорта моделей в различных форматах, что значительно упрощает и ускоряет процесс разработки.

Создание собственной модели нейронной сети на основе ранее собранного датасета позволяет точнее адаптировать модель к конкретным требованиям и особенностям задачи классификации кузовов автомобилей.

Процесс разработки модели включает в себя несколько этапов, включая предварительную обработку данных, обучение модели на тренировочных данных и проверку ее эффективности на тестовых данных.

Библиографический список

1. Звайгзне А.Ю. Разработка классификационной модели видов кузова автомобилей по фотографии // Постулат. 2023 №6 (107).
2. Lin Q. et al. RoboFlow: a data-centric workflow management system for developing AI-enhanced Robots //Conference on Robot Learning. PMLR, 2022. С. 1789-1794.
3. Bisong E., Bisong E. Google colaboratory //Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners. 2019. С. 59-64.
4. Kalliamvakou E. et al. The promises and perils of mining github //Proceedings of the 11th working conference on mining software repositories. 2014. С. 92-101.
5. Wu W. et al. Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image //PloS one. 2021. Т. 16. №. 10. С. e0259283.