

УДК 004.89

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ УПРАВЛЕНИЯ ПРОЦЕССАМИ ВИДЕОКОНТРОЛЯ

В. К. ШЕШОЛКО,
доцент кафедры управления информационными ресурсами

Т. Ф. СТАРОВОЙТОВА,
доцент кафедры управления информационными ресурсами

Академия управления при Президенте Республики Беларусь

И. А. СТАРОВОЙТОВ,
студент кафедры
информационных систем
и технологий БГТУ

В статье представлен процесс обучения модели распознавания объектов с использованием языка программирования Python и инструментов для обучения нейронных сетей. Метод создания модели машинного обучения включает в себя определение целей и задач модели, выбор подходящего алгоритма обучения (в данном случае нейронных сетей), сбор и подготовку набора данных, настройку модели и тестирование на наборе данных. Также анализируются метрики, используемые для оценки точности работы обученной модели, которые подтверждают ее надежность при работе с загружаемой информацией.

Ключевые слова: машинное обучение, нейронные сети, обработка данных, распознавание объектов, Python, Tensorflow.

Для цитирования: Старовойтов И. А., Шешолко В. К., Старовойтова Т. Ф. Нейросетевые технологии управления процессами видеоконтроля // Веснік сувязі, 2024. – № 6/2024 [188]. – С. 53–57.

ВВЕДЕНИЕ

В современном мире наблюдается стремительное развитие технологий искусственного интеллекта (ИИ) на основе нейронных сетей. Они стали неотъемлемой частью многих областей жизни человека, начиная от простых приложений для смартфонов и заканчивая сложными системами управления и контроля. Одной из наиболее перспективных областей применения нейронных сетей является видеоконтроль с функцией распознавания объектов наблюдения.

С технической точки зрения, обучение и внедрение систем распознавания объектов представляют собой комплексную задачу, включающую в себя совокупность алгоритмов и методов, которые опираются на машинное обучение, модели компьютерного зрения, нейронные сети, процесс обучения и тестирования модели на наборе изображений.

ОСНОВНАЯ ЧАСТЬ

Процессы видеоконтроля включают в себя сбор, обработку и анализ видеоданных для мониторинга и контроля определенных областей или ситуаций. Современные системы видеоконтроля могут включать в себя следующие компоненты:

- видеокамеры для сбора видеоданных;
- мониторы для отображения видеоданных

в реальном времени;

- системы видеозаписи для хранения и архивации видеоданных;
- программное обеспечение для обработки и анализа видеоданных.

Создание и использование нейронных сетей для управления процессами видеоконтроля открывает новые возможности для мониторинга и анализа видеоданных, касающихся различных сфер жизни общества. В частности, можно выделить некоторые области применения данных технологий.

1. Управление транспортом. Системы распознавания используются для контроля состояния дорожного покрытия, анализа транспортного трафика с целью его оптимизации, распознавания номерных и дорожных знаков, определения препятствий, пешеходов и дорожно-транспортных происшествий.
2. В промышленном производстве системы распознавания на основе нейронных сетей могут использоваться для контроля качества продукции, определения дефектов изделий, автоматизации процессов сортировки и упаковки продукции, роботизации производственных процессов.
3. Безопасность. Системы распознавания могут использоваться для идентификации разыскиваемых лиц, анализа отпечатков



Рисунок 1. Пример изображения из набора данных VisDrone

пальцев, почерка и других биометрических признаков.

4. Ресурсное планирование. Автоматический видеоконтроль может помочь в оптимизации распределения ресурсов, таких как энергия, вода, транспорт.

Основным источником информации, необходимой для дальнейшего анализа, могут выступать камеры видеонаблюдения, а также спутники, самолеты, вертолеты, дроны, выполняющие процессы наблюдения и передающие в дальнейшем информацию оператору. Эти технологии могут предоставить ценную информацию в реальном времени, которая может быть использована для мониторинга и управления различными аспектами общественной жизни.

Особую актуальность в последнее время приобретает использование дронов в качестве средства мониторинга и контроля. Имея относительно невысокую стоимость, дроны могут эффективно применяться для видеонаблюдения, решая множество задач. Например, в области сельского хозяйства дроны с системой распознавания могут контролировать состояние почвы, урожайность, вовремя определять очаги заболеваемости растений, проводить мониторинг состояния лесного фонда, вовремя выявлять лесные пожары или незаконную вырубку леса. Применение дронов может оказать помощь в поисках и спасении людей, в определении масштабов и последствий чрезвычайных ситуаций. При управлении городской инфраструктурой дроны с системой распознавания могут предоставлять информацию, которую можно использовать при планировании городской застройки, строительстве дорог и мостов и других объектов. Данные, получаемые с дронов, помогут при обнаружении пожаров, происшествий на дорогах, повреждений городской инфраструктуры.

Одним из перспективных методов решения задачи распознавания является использование системы компьютерного зрения YOLO (You Only Look Once) [1], которая позволяет обнаруживать и классифицировать объекты на изображениях или

видео в режиме реального времени. В этой системе входное изображение разбивается на ячейки, для которых определяется наличие объекта, его координаты, размеры и класс. С помощью специального алгоритма отбрасываются несущественные определения объектов, а оставшиеся объединяются в один список. Это позволяет YOLO быстро и эффективно идентифицировать различные объекты за один этап через нейронную сеть.

Обучим модель распознавания объектов на расстоянии с использованием набора данных VisDrone – крупномасштабного датасета, содержащего аннотированные данные изображений, снятых с помощью дронов.

VisDrone состоит из 288 видеоклипов с 261 908 кадрами и 10 209 статичными изображениями, снятыми различными камерами, установленными на дронах. Набор данных охватывает широкий спектр аспектов, включая местоположение (14 различных городов Китая), окружение (городское и сельское), объекты (пешеходы, автомобили, велосипеды и т. д.) и плотность (разреженные и переполненные сцены). Набор данных был собран с помощью различных беспилотных платформ при различных сценариях, погодных условиях и освещении. Эти кадры вручную аннотированы более чем 2,6 млн ограничительных рамок таких объектов, как пешеходы, автомобили, велосипеды и трехколесные велосипеды. Для более эффективного использования данных также предусмотрены такие атрибуты, как видимость сцены, класс объекта и окклюзия. Пример изображения, размещенном в данном датасете, представлен на рис. 1.

Модифицированный подход к получению данных заключается в том, что необработанный набор изображений предварительно нормализован, разделен на набор уменьшенных по размеру элементов, дополнен модификациями уже существующих снимков (поворотом на N градусов по одной из осей). Рассматриваемый датасет будет загружен в память компьютера автоматически в момент запуска процесса обучения модели.

Для анализа информации используется библиотека Ultralytics, содержащая в себе набор методов YOLO, необходимых для организации процесса обучения моделей машинного обучения. Наиболее подходящей версией является набор YOLOv11 [2], актуальный на момент написания статьи, включающий в себя множество архитектурных и технических улучшений по сравнению с предыдущими вариантами. Существует множество различных архитектур для реализации задач в сфере object detection (RetinaNet, Faster R-CNN), однако YOLO выделяется своей изначальной точностью, полученной в процессе предварительного обучения, и простотой внедрения, что делает его оптимальным выбором для решения поставленной задачи.

Для запуска процесса обучения был написан код на языке программирования Python версии 3.10. При этом возможна организация хранения решения как в файле формата PY, так и в блокноте Jupyter. На первом этапе произведен импорт необходимой библиотеки, далее загружается предварительно подготовленная модель. На следующем этапе проводится обучение модели с использованием указанного набора данных в 100 этапов, с размером целевого изображения 640×640 пикселей. Пример кода для загрузки данных и обучения модели представлен на рис. 2.

После выполнения всех процессов, связанных со скачиванием, обработкой данных и обучением модели, получен результат распознавания объектов на пользовательских снимках, ранее не используемых в модели. Пример обработанного изображения после обучения представлен на рис. 3.

В процессе обучения происходит совмещение текстовой информации о нахождении объектов на изображениях со снимком, в результате чего составляются учебные наборы, один из которых представлен на рис. 4.

Результатом процесса обучения является новая нейронная сеть, отличающаяся объединением трех последовательных этапов (подготовка данных, обучение модели и ее экспорт), использование которой возможно для дальнейшего получения отклика с целью распознавания объектов на загружаемых фото.

Далее, с использованием алгоритмов библиотеки Ultralytics выполним подсчет точности распознавания объектов и сохраним данную информацию в общей директории для анализа ее в будущем [3].

Рассматриваемые метрики в совокупности помогают оценить, насколько хорошо модель справляется с задачей классификации. Они учитывают различные аспекты производительности модели и могут быть особенно полезны при работе с несбалансированными наборами данных [4]. График расчета метрики точности распознавания классов объектов на изображении F1 в процессе

```
import torch
from torch.utils.data import DataLoader
from torchvision import datasets, transforms
from ultralytics import YOLO

model = YOLO('yolo11n.pt')

# Определение трансформации данных
transform = transforms.Compose([
    transforms.Resize((640, 640)),
    transforms.ToTensor()
])

# Загрузка набора данных
dataset = datasets.ImageFolder('D:/Data/VisDrone',
                               transform=transform)
dataloader = DataLoader(dataset, batch_size=32,
                        shuffle=True)

criterion = torch.nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)

for epoch in range(100):
    for i, (inputs, labels) in enumerate(dataloader):
        inputs, labels = inputs.to(device),
        labels.to(device)

        outputs = model(inputs)
        loss = criterion(outputs, labels)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        if (i+1) % 100 == 0:
            print(f'Epoch [{epoch+1}/100], Step
                  [{i+1}/{len(dataloader)}], Loss: {loss.item()}')
```

Рисунок 2. Программный код подготовки данных и обучение модели



Рисунок 3. Распознавание объектов в результате обучения модели



Рисунок 4. Вид совмещения «лейблов» и изображений для обучения модели

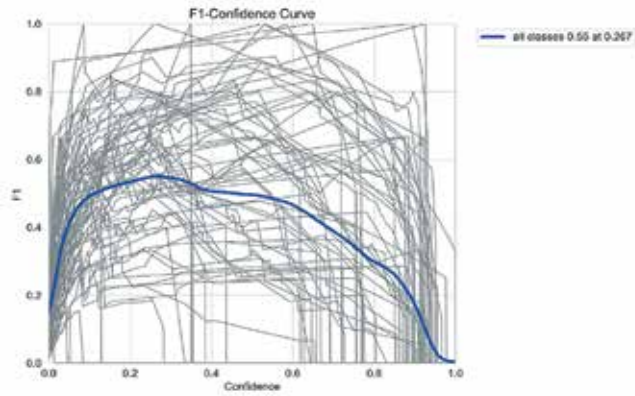


Рисунок 5. F1 модели машинного обучения

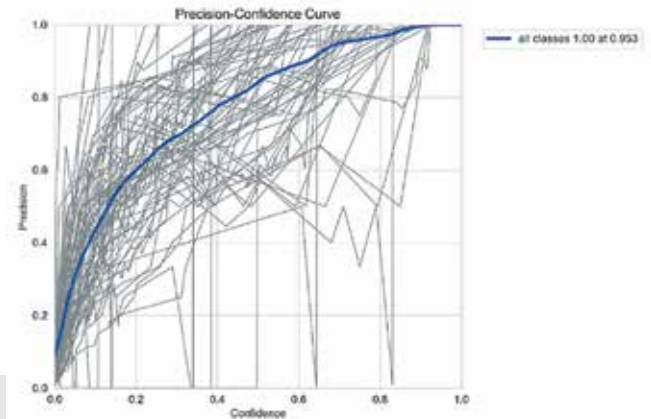


Рисунок 6. Кривая Precision-Confidence модели

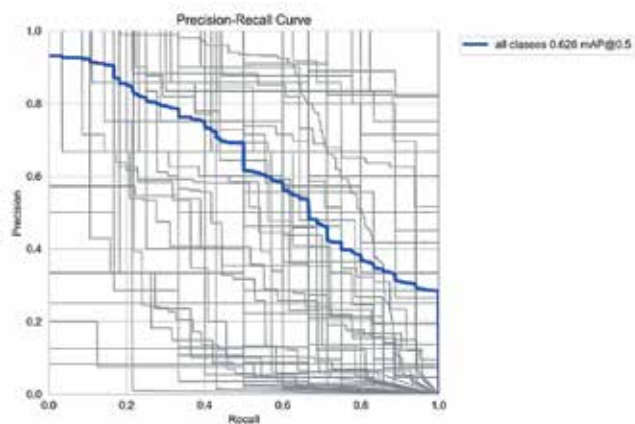


Рисунок 7. Кривая Precision-Recall модели

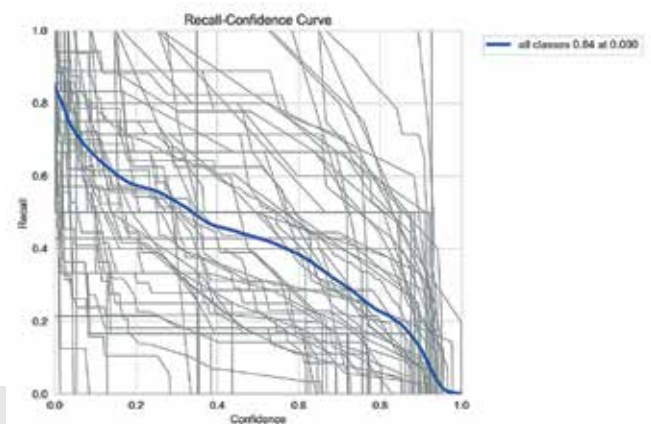


Рисунок 8. Кривая Recall-Confidence модели

обучения представлен на рис. 5. Данный график показывает, что модель точно распознает большинство классов объектов после определенного количества эпох обучения. Это свидетельствует о том, что модель успешно обучается и улучшает свою способность к распознаванию с течением времени.

Кроме того, для анализа также доступно соотношение точности распознавания объекта к уверенности или достоверности модели (Precision-Confidence), представленное на рис. 6. Данный график показывает, что модель уверена в полученных результатах, т. к. большинство точек на графике Precision-Confidence сосредоточены в верхнем правом углу. Это означает, что модель не только точно распознает объекты, но и уверена в своих предсказаниях.

Для оценки основных метрик Precision и Recall была разработана кривая, представленная на рис. 7. Данная кривая отражает процесс уменьшения точности с увеличением полноты. Эта кривая является важным инструментом для анализа производительности модели классификации. Она позволяет наблюдать баланс между точностью и полнотой при различных пороговых значениях.

В идеальном случае требуется достичь 100%-ной точности и 100%-ной полноты, но на практике это часто невозможно из-за присутствия шума и неоднозначности в данных. Поэтому выбор оптимального порога зависит от конкретной задачи и требований к модели. Важно отметить, что увеличение полноты за счет снижения точности может быть приемлемым в некоторых случаях, но не в других. Например, в медицинской диагностике, где пропуск болезни может иметь серьезные последствия, высокая полнота может быть более важной, чем высокая точность. Однако в системах рекомендаций, где пользователь может быть разочарован большим количеством нерелевантных результатов, высокая точность может быть более предпочтительной. Таким образом, кривая Precision-Recall помогает лучше понять компромисс между точностью и полнотой и выбрать наиболее подходящую модель для задачи.

Для оценки соотношения полноты и уверенности модели был создан график, представленный на рис. 8. Данный график показывает соотношение метрик «Полнота» и «Уверенность», позволяет понять степень их зависимости, чтобы в дальнейшем использовать эту информацию для



достижения определенной полноты либо уверенности от обученной модели [5].

Благодаря наличию большого количества информации о процессе обучения имеется возможность своевременно вносить изменения для получения наилучших результатов и снижения вероятности получения некорректных результатов.

ЗАКЛЮЧЕНИЕ

В современных условиях, когда необходимы обработка и анализ большого объема данных, автоматизация распознавания объектов с использованием нейронных сетей становится все более актуальна. Нейронные сети могут выявлять сложные закономерности и точно прогнозировать, что делает их незаменимым инструментом в сфере видеоконтроля. Написание программного кода на языке Python позволило создать нейронную сеть на основе инструмента YOLOv11, реализующего методы, необходимые для автоматизации управления данными. Было использовано обучение модели на основе большого количества

аннотированных изображений, выполнен подсчет точности распознавания объектов, что позволило организовать процессы отслеживания и анализа данных на фото и видеозаписях.

Использование данного инструмента позволяет оптимизировать процесс подготовки данных и обучения модели: необработанный набор изображений предварительно нормализован, разделен на набор уменьшенных по размеру элементов, дополнен модификациями уже существующих снимков (поворотом на N градусов по одной из осей), загружен для обучения.

Представленный цикл обучения нейронной сети включает три последовательных этапа (подготовка данных, обучение модели и ее экспорт) с учетом анализа корректности точности распознавания объекта. Такой подход отличается от существующих методов, не предусматривающих объединение всех трех процессов, реализующих обучение. Это позволяет ускорить процесс обучения и предоставить детальную информацию о его результатах.

ЛИТЕРАТУРА

1. Ultralytics Docs : [сайт] – URL: <https://docs.ultralytics.com/ru/> (дата обращения: 05.05.2024).
2. Метрики классификации и регрессии [сайт] – URL: <https://education.yandex.ru/handbook/ml/article/metriki-klassifikacii-i-regressii> (дата обращения: 05.05.2024).
3. Метрики в задачах машинного обучения [сайт] – URL: <https://habr.com/ru/companies/ods/articles/328372/> (дата обращения: 05.05.2024).
4. Metrics and scoring: quantifying the quality of predictions : [сайт] – URL: https://scikit-learn.org/stable/modules/model_evaluation.html (дата обращения: 05.05.2024).
5. Train YOLOv8 on Custom Dataset – A Complete Tutorial [Сайт] – URL: <https://learnopencv.com/train-yolov8-on-custom-dataset/> (дата обращения: 05.05.2024).

The article develops an object recognition model using the Python programming language and neural network training tools. The method of creating a machine learning model includes defining the goals and objectives of the model, selecting a suitable training algorithm (in this case, neural networks), collecting and preparing a data set, tuning the model and testing it on the data set. Metrics used to assess the accuracy of the trained model are also analyzed, confirming its reliability when working with uploaded information.

Key words: machine learning, neural networks, data processing, object recognition, Python, Tensorflow.

For citation: Starovoitov I. A., Shesholko V. K., Starovoitova T. F. Neural network technologies of video control processes management // Vesnik suvyazi. 2024. – № 6/2024 (188). – С. 53–57.