

В данной статье для исследования зависимости использовались два метода подготовки обучающего набора данных.

В первом методе используется высокая детализация разметки изображений (рис. 2). Оператор, выступающий в роли учителя, в таком случае выделяет даже самые «мелкие», едва заметные, объекты (под «мелкими» объектами подразумеваются такие объекты, которые состоят из небольшого количества пикселей, по сравнению с размером изображения). Таким образом, достигается наиболее полное множество возможных выделенных объектов на изображениях.



Рисунок 2 – Разметка с высокой детализацией

Во втором методе используется умеренная детализация разметки изображений (рис. 3). Здесь, оператор выделяет только те объекты, которые он может абсолютно точно классифицировать, а также, которые при более детальном рассмотрении не размываются и, тем самым, не уменьшают уверенности оператора в выбранном классе для этого объекта. Такие объекты, как правило, занимают довольно обширную площадь по отношению к площади всего изображения.

Между описанными выше методами разметки нет точно измеримой границы, которая определяет, что может считаться более детализированной или менее детализированной разметкой, но тем не менее, она есть и на приведенных, в качестве примера, рисунках данная граница отчетливо видна.



Рисунок 3 – Разметка с умеренной детализацией

Используя два вида разметки с различной детализацией, было подготовлено два обучающих набора данных по 3000 изображений в каждом.

Для обучения искусственной нейронной сети был написан программный код на языке Python версии 3 с использованием библиотеки для создания искусственных нейронных сетей с возможностью глубокого обучения ImageAI. Данная библиотека позволяет проводить глубокое обучение, при этом от программиста требуется использование всего лишь нескольких функций данной библиотеки, что в совокупности может занимать несколько строк кода. Программный код, который использовался для обучения нейронной сети в данной статье приведен на рис. 4.

```
1 from imageai.Detection.Custom import DetectionModelTrainer
2
3 trainer = DetectionModelTrainer()
4 trainer.setModelTypeAsYOLOv3()
5 trainer.setDataDirectory(data_directory = "kitti")
6 trainer.setTrainConfig(
7     object_names_array = ["car", "truck", "bus", "train", "person"],
8     batch_size = 4, num_experiments = 30,
9     train_from_pretrained_model = "pretrained-yolov3.h5")
10 trainer.trainModel()
```

Рисунок 4 – Программный код для обучения нейронной сети

Обучение проводилось на уже предобученной нейронной сети, которая была взята с сайта библиотеки ImageAI. Принцип обучающего алгоритма, следующий: на вход подается уже предобученная нейронная сеть и набор подготовленных изображений (в рамках данной статьи размер набора – 3000 изображений). Затем алгоритм выполняет 30 итераций обучения (или их ещё называют «эпохи», «epoch»), при этом на каждой итерации происходит сохранение в файл обученной модели искусственной нейронной сети с полученными весами. На каждом этапе алгоритма вычисляется точность обучения нейронной сети и выводится в консоль.

Обучение производилось на компьютере со следующими характеристиками:

1. процессор – Intel Core i7-8700K 3.70GHz;
2. оперативная память – 32 ГБ;
3. видеокарта – NVIDIA GeForce GTX 1080Ti.

В итоге, обучение нейронных сетей на двух разных наборах данных прошло успешно. В среднем, для каждого набора данных обучение заняло около 17 часов. Выведенная информация в консоль по завершении обучения показана на рис. 5 для первого набора данных и на рис. 6, для второго.

```
Epoch 1/30
3000/3000 [=====] - 2020s 673ms/step -
loss: 35.1501 - yolo_layer_1_loss: 3.3680 - yolo_layer_2_loss:
8.9870 - yolo_layer_3_loss: 22.7951 - val_loss: 14.8710 -
val_yolo_layer_1_loss: 2.1580 - val_yolo_layer_2_loss: 6.0250 -
val_yolo_layer_3_loss: 6.6880
Epoch 2/30
3000/3000 [=====] - 2031s 677ms/step -
loss: 11.6791 - yolo_layer_1_loss: 1.325 - yolo_layer_2_loss:
4.4778 - yolo_layer_3_loss: 5.8763 - val_loss: 7.4578 -
val_yolo_layer_1_loss: 1.6543 - val_yolo_layer_2_loss: 2.359 -
val_yolo_layer_3_loss: 3.4445
...
Epoch 30/30
3000/3000 [=====] - 2012s 670ms/step -
loss: 4.5123 - yolo_layer_1_loss: 0.9122 - yolo_layer_2_loss:
1.5412 - yolo_layer_3_loss: 2.0589 - val_loss: 5.8369 -
val_yolo_layer_1_loss: 1.3671 - val_yolo_layer_2_loss: 2.0157 -
val_yolo_layer_3_loss: 2.4541
```

Рисунок 5 – Лог консоли по завершении обучения на первом наборе данных

```
Epoch 1/30
3000/3000 [=====] - 2010s 670ms/step -
loss: 33.2678 - yolo_layer_1_loss: 5.9850 - yolo_layer_2_loss:
11.4578 - yolo_layer_3_loss: 15.8250 - val_loss: 13.5489 -
val_yolo_layer_1_loss: 2.3654 - val_yolo_layer_2_loss: 5.1289 -
val_yolo_layer_3_loss: 6.0546
Epoch 2/30
3000/3000 [=====] - 1984s 661ms/step -
loss: 10.1259 - yolo_layer_1_loss: 1.5681 - yolo_layer_2_loss:
4.1853 - yolo_layer_3_loss: 4.3725 - val_loss: 6.9633 -
val_yolo_layer_1_loss: 1.3840 - val_yolo_layer_2_loss: 2.6421 -
val_yolo_layer_3_loss: 2.9372
...
Epoch 30/30
3000/3000 [=====] - 1993s 664ms/step -
loss: 2.5460 - yolo_layer_1_loss: 0.4891 - yolo_layer_2_loss:
0.7136 - yolo_layer_3_loss: 1.3433 - val_loss: 3.1435 -
val_yolo_layer_1_loss: 0.7743 - val_yolo_layer_2_loss: 1.0742 -
val_yolo_layer_3_loss: 1.2950
```

Рисунок 6 – Лог консоли по завершении обучения на втором наборе данных

В итоге, с помощью библиотеки ImageAI и предобученной нейронной сети удалось провести глубокое обучение и получить две сверточные нейронные сети, обученные на двух разных наборах данных. Суммарная ошибка обучения в первом случае оказалась равна 5.8369, а во втором случае – 3.1435. Также можно проследить, что на протяжении всего обучения, начиная с первой итерации, суммарная ошибка для второго набора данных была ниже, чем для первого набора данных. Этот и предыдущий факт подтверждают предположение, что детализация при подготовке обучающих данных влияет на суммарную ошибку обучения.

Наличие такой зависимости может объясняться влиянием нескольких факторов. Во-первых – это человеческий фактор. Размечая объекты небольшого размера, вероятность совершить ошибку увеличивается, поскольку отношение области, в случае совершения ошибки, которая не относится к выделяемому объекту, к области самого объекта гораздо выше, чем для больших объектов на изображениях. Другой фактор – это программное обеспечение, посредством которого происходит разметка. В ряде программ, специализирующихся на разметке изображений, имеется проблема при разметке мелких объектов – это проблема точности. При выделении небольших объектов не представляется возможным выделить их достаточно точно. Путем анализа выявлено, что эта проблема имеет место при сохранении разметки в текстовый файл, поскольку после повторного открытия явным образом наблюдается смещение выделенных рамок.

При обучении нейронной сети на наборе изображений, полученных первым методом, происходит, соответственно, недостаточно верная корректировка весов модели, поскольку качество подготовленных данных напрямую влияет на ошибку обучения и погрешность обучаемой нейронной сети. При обучении на втором наборе данных, влияние неточности разметки изображений намного ниже, что позволяет добиться более лучшего результата.

Таким образом, использование полученных результатов данного исследования позволит значительно ускорить подготовку обучающего набора данных и уменьшить суммарную ошибку обучения искусственной нейронной сети.

Литература

1. Чуйков Р.Ю., Юдин Д.А. Обнаружение транспортных средств на изображениях загородных шоссе на основе метода Single shot multibox detector // Научный результат. Информационные технологии. – 2017. – Т.2, №4. – С. 50–58.
2. Кафтанников И.Л., Парасич А.В. Проблемы формирования обучающей выборки в задачах машинного обучения // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2016. – Т. 16, № 3. – С. 15–24.
3. Рындин А.А., Ульев В.П. Исследование скорости обучения нейронных сетей // Вестник ВГТУ. – 2012. – Т. 5, № 8. – С. 7–9.
4. Татьянкин В.М., Дюбко И.С. Обучающая выборка в задаче распознавания образов при использовании нейронных сетей // Вестник Югорского государственного университета. – 2015. – №. 2 (37). – С. 94–98.
5. Бредихин А.И. Алгоритмы обучения сверточных нейронных сетей // Вестник Югорского государственного университета. – 2019. – №. 1 (52). – С. 41–54.
6. Азарченков А.А., Любимов М.С., Лушков В.И. Распознавание объектов дорожной инфраструктуры с помощью полносверточной нейронной сети // Автоматизация и моделирование в проектировании и управлении. – 2019. – №4 (6). – С.38–43

