

Обнаружение повреждений дорожного покрытия в видеопотоке автомобильного видеорегилятора

И.А. Мисюрина

*Самарский национальный исследовательский университет
им. академика С.П. Королева
Самара, Россия
ira.misyurina.smr@gmail.com*

П.Ю. Якимов

*Самарский национальный исследовательский университет
им. академика С.П. Королева
Самара, Россия
pavel.y.yakimov@gmail.com*

Аннотация – В настоящей работе описывается разработка программного обеспечения для детекции и классификации повреждений дорожного покрытия различных типов. Описывается алгоритм работы YOLO, дано краткое описание выбранных для разработки моделей. Описана архитектура реализованного ПО.

Ключевые слова – Нейронная сеть, детекция, классификация, техническое обслуживание дорог, YOLO, YOLOv5, CNN.

1. ВВЕДЕНИЕ

Исследования по обнаружению повреждений дорожного покрытия с использованием методов обработки изображений и машинного обучения представляют интерес как для развитых, так для развивающихся стран. Актуальность этого вопроса обусловлена тем, что дороги являются одной из важнейших гражданских инфраструктур в каждой стране, они вносят прямой или косвенный вклад в экономику государств и, что более важно, в благосостояние и безопасность граждан. Таким образом, техническое обслуживание дорог имеет высокий приоритет, и многие страны стремятся внедрять новейшие механизмы и стандарты инспекции для осуществления этого процесса.

Традиционный способ обнаружения повреждений дорожного покрытия с применением машинного обучения – детекторы на основе фильтров (например, фильтр Габора [1]) и системы использующие созданные вручную признаки [2, 3]. Первые попытки использования нейросетей в данной области были предприняты в 2015-2016 гг. исследователями из США, собравшими набор данных и обучившими свёрточную нейронную сеть распознавать трещины на дороге [4].

С тех пор был проведен ряд исследований, посвященных сравнению эффективности архитектур сетей при распознавании повреждений разных типов [5 - 7], также было разработано несколько полноценных детекторов состояния дорог на основе нейронных сетей [8, 9].

Целью данной работы является разработка программного продукта, позволяющего детектировать и классифицировать повреждения дорожного покрытия при помощи нейронных сетей.

2. ОПИСАНИЕ МОДЕЛЕЙ

Стандартным способом решения задачи детекции и классификации является разбиение изображения на области с последующей классификацией областей на наличие объекта, то есть изображение просматривается дважды, в первый раз во время выявления областей, во

второй – во время классификации объекта. Одноступенчатые детекторы, такие как YOLO, используют иной подход: из исходного изображения формируется квадратная матрица, в каждой клетке которой записана информация о наличии объекта определенного класса в соответствующей данной клетке части изображения. Алгоритм позволяет получить всю необходимую информацию за один раз, что существенно увеличивает скорость обработки [10].

Поскольку одним из ключевых факторов выбора моделей являются требования к ресурсоемкости и скорости, двухступенчатые детекторы (R-CNN, Faster R-CNN) не рассматривались как оптимальные решения. Среди одноступенчатых детекторов выбор происходил между SSD, YOLOv3, YOLOv4 и YOLOv5. В итоге в качестве основы для работы была выбрана пятая версия YOLO, поскольку, согласно результатам оценки на наборе данных COCO, модели этой архитектуры не уступают остальным в точности и превосходят их в скорости работы [11].

Для создания программы использовано четыре модели, все они относятся к пятому поколению YOLO и полностью реализованы на Pytorch, две основаны на архитектуре пятой версии, две – на архитектуре третьей версии.

В качестве первой модели выбрана стандартная версия YOLOv5s, предоставленная разработчиками. Модель 2 составлена самостоятельно на основе архитектуры YOLOv5, основные отличия касаются backbone – добавлена еще пара слоев свертки и слой-bottleneck, также вместо модуля C3 (содержащего 3 слоя Conv2d) использованы слои BottleneckCSP (содержащие в общей сложности 4 слоя Conv2d). Модели 3 и 4 основаны на архитектуре YOLOv3. Третья модель повторяет архитектуру YOLOv3-SPP, с заменой оригинальных модулей объединения на C3. Четвертая модель использует модули BottleneckCSP. YOLOv3-SPP выбрана за основу как модель с лучшей точностью (average precision) в семействе YOLOv3.

3. ЭКСПЕРИМЕНТ

Для обучения моделей, краткое описание архитектуры которых приведено в предыдущем разделе, использован набор данных, включающий 4 типа повреждений:

- D00 – единичная продольная трещина;
- D10 – единичная поперечная трещина;
- D20 – множество трещин мозаичного типа;
- D40 – выбоина, яма.

После обработки объем набора данных составлял порядка 12 000 фото, данные были разделены на тестовую и обучающую выборки в пропорции 2:8. Обучение производилось в Google Colab на Tesla T4, в качестве основной метрики для оценки моделей использована mAP (mean Average Precision). На Рис. 1 изображены предсказания модели, полученные в процессе обучения.



Рис. 1. Пример предсказаний модели

Все модели обучались 100 эпох, соответственно, лучшие веса, полученные за это время, стали итоговыми версиями моделей. Результаты эксперимента представлены в Таблице 1.

Таблица 1. ЗНАЧЕНИЯ mAP_{0.5} ДЛЯ МОДЕЛЕЙ

Класс	Модель 1	Модель 2	Модель 3	Модель 4
Все	0,580	0,556	0,582	0,577
D00	0,561	0,535	0,571	0,559
D10	0,464	0,411	0,458	0,455
D20	0,684	0,710	0,689	0,685
D40	0,610	0,567	0,609	0,611

Результаты, полученные в ходе экспериментальных исследований демонстрируют точность, близкую к точности, полученной участниками Global Road Damage Detection Challenge, решавшими схожую задачу [12]. В целях повышения точности планируется рассмотреть сети других архитектур, в частности Mask R-CNN, а также сделать соответствующую разметку набора данных.

4. ИТОГОВЫЙ ПРОГРАММНЫЙ МОДУЛЬ

Архитектурно реализованная программа представляет собой REST-сервис. Для работы доступны все обученные модели. Серверная часть системы состоит из двух независимых Flask-модулей, один из которых отвечает за обработку фото, другой – за обработку видеоданных, в качестве application-сервера в обоих случаях используется Gunicorn. Клиентская часть написана на React, сервер – Nginx.

На данный момент система находится на стадии тестирования. Интерфейс сервиса представлен на Рис. 2.

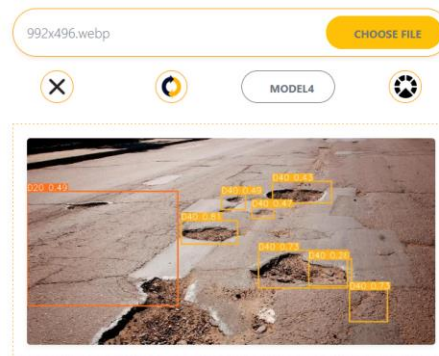


Рис. 2. Интерфейс сервиса

5. ЗАКЛЮЧЕНИЕ

В результате проделанной работы была реализована альфа-версия разрабатываемой программной системы. Спроектированы и обучены модели для детекции и классификации повреждений дорожного покрытия. В дальнейшем планируется собрать более релевантный обучающий набор с целью повышения качества работы алгоритма на данных с российских автодорог. Разработанный прототип сервиса запущен на тестовом стенде и готов к внедрению.

ЛИТЕРАТУРА

- [1] Salman, M. Pavement crack detection using the Gabor filter / M. Salman, S. Mathavan, K. Kamal, M. Rahman // 16th International IEEE Conference on Intelligent Transportation Systems (ITSC). – 2013. – P. 2039–2044.
- [2] Hu, Y. A novel LBP based methods for pavement crack detection / Y. Hu, C. Zhao // Journal of Pattern Recognition Research. – 2010. – Vol. 5(1). – P. 140-147.
- [3] Akarsu B. A fast and adaptive road defect detection approach using computer vision with real time implementation / B. Akarsu , M. Karaköse , K. Parlak , E. Akin, A. Sarimaden // International Journal of Applied Mathematics Electronics and Computers. – 2016. –Vol. 1(1). – P. 290-295.
- [4] Zhang, L. Road crack detection using deep convolutional neural network / L. Zhang, F. Yang, Y.D. Zhang, Y.J. Zhu // 2016 IEEE International Conference on Image Processing (ICIP). – 2016. – P. 3708-3712.
- [5] Ale, L. & Zhang, Ning & Li, Longzhuang. Road damage detection using RetinaNet / L. Ale, N. Zhang, L. Li // 2018 IEEE International Conference on Big Data (Big Data). – 2018. – P. 5197-5200.
- [6] Maeda, H. Road damage detection and classification using deep neural networks with smartphone images / H. Maeda Y. Sekimoto, T. Seto, T. Kashiyama, H. Omata // Computer - Aided Civil and Infrastructure Engineering. – 2018. – Vol. 33 (12). – P. 1127-1141.
- [7] Jeong, D. Road damage detection using YOLO with smartphone images / D. Jeong // 2020 IEEE International Conference on Big Data (Big Data). – 2020. – P. 5559-5562.
- [8] Maeda, H. Generative Adversarial Network for road damage detection / H. Maeda Y. Sekimoto, T. Seto, T. Kashiyama, H. Omata // Computer - Aided Civil and Infrastructure Engineering. – 2021. – Vol. 36(1). – P. 47-60.
- [9] Maeda, H. Lightweight road manager: smartphone-based automatic determination of road damage status by deep neural network / H. Maeda, Y. Sekimoto, T. Seto // Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems. – 2016. – P. 37-45.
- [10] Redmon J. You only look once: Unified, real-time object detection / J. Redmon, S. Divvala, R. Girshick, A. Farhadi // Proceedings of the IEEE conference on computer vision and pattern recognition. – 2016. – P. 779-788.
- [11] Github [Electronic resource]. — Access mode: <https://github.com/ultralytics/yolov5#readm/> (01.02.2023)
- [12] Arya, D. Global road damage detection: State-of-the-art solutions / D. Arya, H. Maeda, S. Ghosh, D. Toshniwal, H. Omata, T. Kashiyama, Y. Sekimoto // 2020 IEEE International Conference on Big Data (Big Data). – 2020. – P. 5533-5539.