

тивную память для хранения токенов, построения очередей запросов, а также для кэширования данных. Каждый из наших модулей имеет доступ к Redis, это сделано для того, чтобы отдельные сервисы могли проверить валидность токена, который приходит вместе с запросом, т.к. наши сервисы – отдельные приложения, которые ничего не знают друг о друге и для проверки авторизации они используют хранилище Redis для поиска токена, который был записан модулем авторизации. Данный подход сокращает риски при разработке приложений и тем самым ускоряет ее.

**Д. А. Кирпичев, К. С. Курочка**  
(ГГТУ им. П. О. Сухого, Гомель)

## **ИСПОЛЬЗОВАНИЕ РЕКУРРЕНТНЫХ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧЕ РАСПОЗНАВАНИЯ АВТОМОБИЛЕЙ НА ИЗОБРАЖЕНИИ И ПОДСЧЁТА ИХ КОЛИЧЕСТВА**

При распознавании объектов на изображении с помощью нейронных сетей важно правильно выбрать архитектуру нейронной сети.

Рассмотрим две структуры нейронных сетей для детектирования объектов на изображении. Первая архитектура – *R-CNN*. Принцип её работы заключается в подаче не всего изображения целиком, а предварительно выделенных другим алгоритмом регионов, на которых предположительно имеются какие-то объекты. Для обработки частей изображения используется алгоритм *Selective Search*. Вторая архитектура – *YOLO*. Главная особенность этой архитектуры по сравнению с первой состоит в том, что первая архитектура применяет *CNN* несколько раз к разным регионам изображения, в *YOLO CNN* применяется один раз ко всему изображению сразу. Сеть делит изображение на своеобразную сетку и предсказывает *bounding boxes* и вероятности того, что там есть необходимый объект для каждого участка сетки.

При сравнении этих архитектур можно сделать вывод о том, что скорость работы второй сети выше, чем первой, а точность детектирования объектов выше у первой при прочих равных условиях.

Для задачи распознавания автомобилей была выбрана *YOLO* архитектура. Результат обучения сети представлен на рисунке 1.

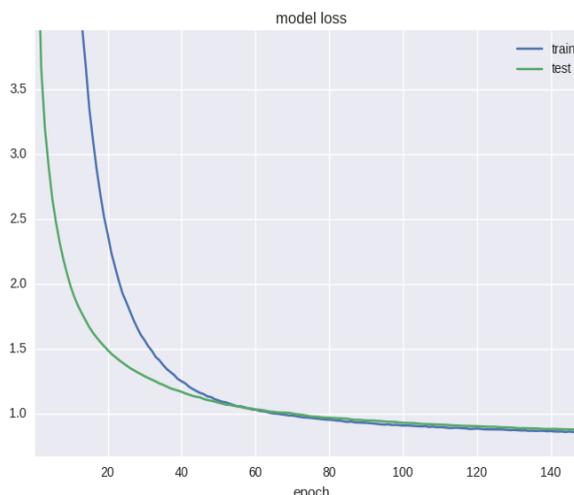


Рисунок 1 – График минимизации функции ошибки в ходе обучения сети

**М. А. Климович, А. Ю. Кульша**  
(БГУИР, Минск)

## **РАЗРАБОТКА И ПРОГРАММИРОВАНИЕ ХОДОВОЙ ЧАСТИ РОБОТА НА ОСНОВЕ ОМНИ-КОЛЕС**

При разработке любого робота ставиться вопрос о том, каким способ будет реализовано движение робота. Одним из вариантов для ходовой части является использование омни-колес. Омни-колеса позволяют роботу двигаться в любом направлении не меняя ориентацию в пространстве, при этом трение скольжения между колесами и поверхностью практически отсутствует. Из минусов стоит отметить, что механизм не может находиться на наклонной поверхности без использования специального стояночного механизма из-за маленького трения.

Для примера разберем вариант, где робот имеет 4 омни-колеса. Для того, чтобы платформа ехала в направлении движения, противоположные колеса должны иметь одно направление в пространстве, тогда исходя из этого можно перенести движение на двухмерную плоскость, задавая движение суммой векторов  $x$  и  $y$ . Пример на рисунке 1.

Для вращения робота вокруг своей оси требуется, чтобы противоположные колеса двигались в разные направления, имея одинаковую скорость.