

КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ
ИМЕНИ АЛЬ-ФАРАБИ

ЛЕКЦИЯ № 5

Обучение нейронных моделей для распознавания транспортных потоков средствами компьютерного зрения. Интеллектуальное управление светофорами

Бельгибаев Бауржан Абдрахимович
Д.т.н., доцент · Кафедра «Искусственный интеллект и big data»
«Технологии в интеллектуальных системах управления в IoT» · 4 курс, бакалавриат

1 Цель лекции

Сформировать системное представление о построении интеллектуальной IoT-системы управления светофорным объектом, использующей видеоданные, нейросетевое обнаружение и подсчёт транспортных средств, оценку состояния очередей и алгоритмы выбора фаз при соблюдении требований безопасности движения.

2 План лекции

1. Транспортный перекрёсток как объект управления и источники данных
2. Архитектура компьютерного зрения и IoT для умного светофора
3. Формирование и разметка обучающего набора данных
4. Обучение, проверка и внедрение нейросетевой модели обнаружения
5. Подсчёт транспорта и оценка интенсивности, очередей и задержек
6. Алгоритмы интеллектуального светофорного регулирования
7. Практический расчёт, оценка качества и ограничения безопасности

1 Перекрёсток как динамический объект управления

Городской перекрёсток представляет собой дискретно-непрерывную динамическую систему: транспортные потоки изменяются во времени, а сигнальные группы светофора переключаются между конечным набором фаз. Цель управления — уменьшить суммарную задержку, длину очередей и число остановок, не нарушая правил конфликтности транспортных направлений и минимально допустимых интервалов безопасности.

Традиционный контроллер реализует заранее заданные циклы и длительности фаз. Адаптивный контроллер использует измерения с детекторов, а интеллектуальный — дополнительно результаты компьютерного зрения, прогнозы и алгоритмы оптимизации. Важно различать оценку количества машин в кадре и измерение реального потока: первое является мгновенной характеристикой, второе определяется пересечениями контрольной линии за заданное время.

Введём интенсивность прибытия $\lambda_i(t)$, пропускную способность $\mu_i(t)$, длину очереди $q_i(t)$ и управляющий сигнал $u_i(t)$, определяющий разрешённое движение на направлении i. В простейшей модели сохранения потока: $dq_i/dt = \lambda_i(t) - \mu_i(t)u_i(t)$, при $q_i \geq 0$. На практике скорость обслуживания зависит от насыщенного потока, геометрии, пешеходных переходов и помех.

2 Архитектура интеллектуального светофора с IoT

Типовая цепочка: IP/USB-камера → Raspberry Pi или другой edge-компьютер → предварительная обработка кадров → нейросетевой детектор → трекинг объектов → расчёт показателей потока → модуль принятия решения → промышленный светофорный контроллер. IoT-шлюз передаёт телеметрию оператору через MQTT/HTTPS и принимает допустимые параметры конфигурации. Внешняя аналитика не должна напрямую обходить сертифицированные защитные функции светофорного контроллера.

2.1. Уровни системы

2 Архитектура интеллектуального светофора с IoT (продолжение)

Уровень	Основные функции	Пример компонентов
Сенсорный	Получение видеокадров и временных меток	IP-камера, USB-камера
Пограничные вычисления	Детекция, трекинг, оценка очередей	Raspberry Pi + AI-ускоритель
Управление	Проверка ограничений, выбор разрешённой фазы	Контроллер светофорного объекта
Связь и диспетчеризация	Мониторинг, журналирование, диагностика	MQTT, защищённая сеть, SCADA

2 Архитектура интеллектуального светофора с IoT (продолжение)

Для системы реального времени критична сквозная задержка T_total = T_capture + T_preprocess + T_infer + T_track + T_decision + T_comm. Если обработка кадра занимает 150 мс, номинальная пропускная способность одного последовательного конвейера составляет не более 1/0,15 ≈ 6,7 кадра/с без учёта дополнительных операций. Для управления фазами важнее устойчивость агрегированных измерений за несколько секунд, чем максимальная частота кадров сама по себе.

4 Интеллектуальный перекрёсток

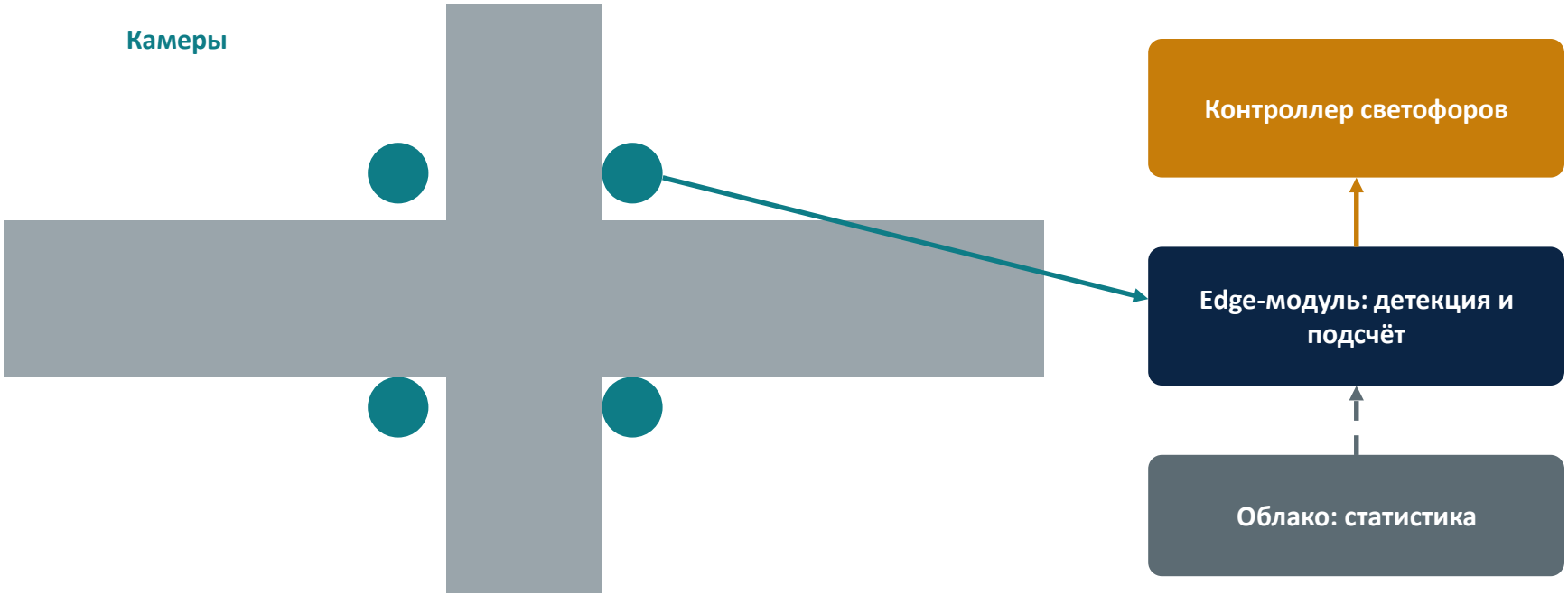


Рис. 8. Камеры на подходах дают видеопоток; edge-модуль формирует данные для регулирования.

3 Подготовка обучающих данных

Набор данных должен включать реальные условия перекрёстка: дневное и ночное освещение, дождь, снег, встречную засветку, плотные очереди, частичные перекрытия объектов и разные типы транспортных средств. Для обнаружения используют ограничивающие прямоугольники (bounding boxes) с метками классов: легковой автомобиль, автобус, грузовик, мотоцикл и др. Для анализа потока также размечают области интереса (ROI), виртуальные линии пересечения и направления движения.

Разделение train/validation/test выполняют по видеозаписям, камерам или датам, а не случайным соседним кадрам: близкие кадры почти одинаковы и создают утечку данных. Важно документировать происхождение видеоматериалов, разрешения на их использование, правила хранения и удаления персональных данных.

3.1. Аугментация

Допустимы изменения яркости, контраста, масштаба, шумов и умеренные геометрические преобразования. Следует избегать трансформаций, искажающих смысл дорожной сцены или разметку направления движения. Дисбаланс классов устраняют сбором дополнительных примеров, взвешиванием потерь или выборкой с контролируемыми пропорциями.

4 Нейросетевое обнаружение транспортных средств

Для edge-устройств часто используют однопроходные детекторы семейства YOLO или компактные модели SSD. Детектор возвращает класс c_j, уверенность p_j и координаты прямоугольника $b_j = (x_j, y_j, w_j, h_j)$. После пороговой фильтрации и подавления дублирующих прямоугольников (NMS) получают набор кандидатов на каждом кадре.

Обучение обычно включает: выбор архитектуры и предварительно обученных весов; приведение разметки к требуемому формату; настройку размера входа, batch size, скорости обучения и эпох; контроль метрик на validation; выбор лучшей контрольной точки; независимую проверку на test; экспорт модели в совместимый формат ускорителя. Необходимо проверить совместимость операций и точности вычислений с конкретным аппаратным устройством.

4.1. Метрики детекции

$Precision = TP/(TP+FP)$; $Recall = TP/(TP+FN)$; $F1 = 2 \cdot Precision \cdot Recall/(Precision+Recall)$. $IoU = |B_{pred} \cap B_{true}| / |B_{pred} \cup B_{true}|$. Средняя точность mAP усредняет качество по классам и заданным порогам IoU. Высокая mAP не гарантирует правильного подсчёта потока, поэтому подсчёт оценивают отдельно.

Пример: TP=180, FP=20, FN=30. Тогда Precision=0,900, Recall=0,857, F1≈0,878. Эти показатели относятся к условному тестовому примеру, а не к измеренным результатам конкретного устройства.

5 От детекции к подсчёту и оценке очередей

При прямом суммировании детекций по кадрам одна и та же машина будет посчитана многократно. Поэтому необходим трекер, связывающий обнаружения между последовательными кадрами и назначающий каждому объекту временный идентификатор. Практически применяются методы сопоставления по геометрии, фильтрация Калмана и алгоритмы SORT/ByteTrack-подобного типа.

Событие проезда фиксируется, когда траектория центра объекта впервые пересекает виртуальную контрольную линию в заданном направлении. Счётчик увеличивается один раз для каждого устойчивого track ID. При смене идентификатора возможен повторный подсчёт; при окклюзии — пропуск события. Для оценки очереди выделяют полосу движения и анализируют стоящие или медленно движущиеся объекты.

Интенсивность потока: $Q_i = 3600 \cdot N_i / \Delta t$, где N_i — число пересечений за окно Δt в секундах, Q_i — автомобилей в час. При $N_i=24$ за $\Delta t=120$ с получаем $Q_i=720$ авт./ч. Средняя задержка: $\bar{d} = (1/N) \Sigma_j d_j$; длина очереди может оцениваться числом стоящих машин или расстоянием от стоп-линии до последнего автомобиля.

5.1. Фильтрация показателей

Для подавления скачков измерений используют экспоненциальное сглаживание: $\hat{q}_i(k)=\alpha q_i(k)+(1-\alpha)\hat{q}_i(k-1)$, $0<\alpha\leq 1$. При $\alpha=0{,}3$, текущей оценке $q=12$ и предыдущей сглаженной $\hat{q}=8$ получаем $\hat{q}_{new}=9{,}2$. Слишком сильное сглаживание повышает инерционность реакции контроллера.

6 Алгоритмы интеллектуального регулирования

6.1. Фиксированный цикл и адаптивное продление зелёной фазы

В фиксированном режиме фазы следуют заранее определённому расписанию. В адаптивном режиме система может продлить текущую зелёную фазу при наличии устойчивого потока, но только до максимальной разрешённой длительности. Минимальная зелёная фаза, межзелёные интервалы, жёлтый сигнал и периоды очистки перекрёстка задаются проектными требованиями и не оптимизируются нейросетью произвольно.

6.2. Выбор фазы по нагрузке

Для двух несовместимых направлений A и B определим вес $W_i = a \cdot \hat{q}_i + b \cdot w_i + c \cdot r_i$, где $\hat{q}_i$ — очередь, w_i — накопленное ожидание, r_i — оценка прибытия в ближайшем окне. Выбор фазы: $i^* = \operatorname{argmax}_i W_i$ среди допустимых фаз. Чтобы исключить голодание малонагруженного направления, вводят максимальное время ожидания и приоритетные правила обслуживания пешеходов и общественного транспорта.

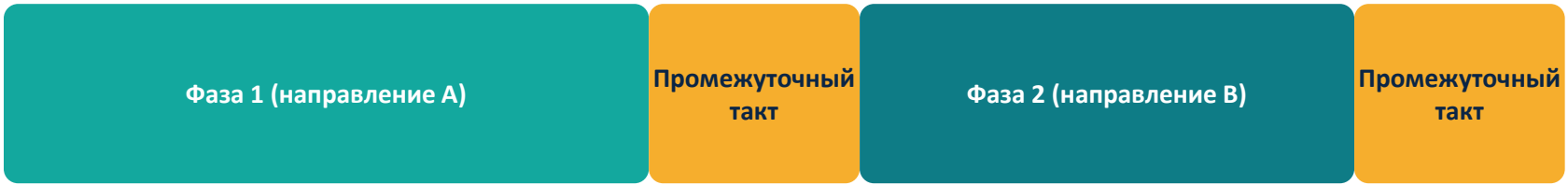
Алгоритмы интеллектуального регулирования (продолжение)

Один из вариантов оптимизационной постановки: минимизировать $J = \Sigma_k \Sigma_i [w_q q_i(k)^2 + w_d d_i(k)] + w_s N_{switch}$, где N_{switch} — число переключений. Ограничения: $g_{min} \le g_i \le g_{max}$; совместимость направлений; обязательные интервалы безопасности; ограничения по пешеходным фазам. Нейронная сеть оценивает состояние или прогноз, а не отменяет ограничения.

6.3. Прогнозирование и интеллектуальные методы

Прогноз прибытия машин на 1–5 минут может выполняться регрессионной моделью, рекуррентной сетью или временной сверточной сетью. Для исследовательского прототипа допустимо сравнить пороговый алгоритм, нечёткий регулятор и модель прогнозирующего управления (MPC). Обучение с подкреплением требует моделирования редких опасных ситуаций и отдельного слоя гарантированно безопасного исполнения.

8 Цикл светофорного регулирования



$$C = \Sigma (g_i + y_i)$$

g_i — длительность зелёного, y_i — промежуточный такт; адаптивный алгоритм перераспределяет g_i по измеренной интенсивности.

Рис. 9. Цикл C складывается из зелёных фаз и промежуточных тактов.

7 Расчётный пример: перекрёсток с двумя фазами

Рассмотрим условный перекрёсток с двумя конфликтующими потоками. По данным компьютерного зрения: $q_A=18$ автомобилей, $q_B=8$ автомобилей; накопленное ожидание $w_A=35$ с, $w_B=55$ с. Примем веса $W_i = q_i + 0{,}1 \cdot w_i$. Тогда $W_A=21{,}5$, $W_B=13{,}5$. Алгоритм предпочёл бы фазу A, если все обязательные ограничения безопасности и минимального обслуживания B выполнены.

Пусть при разрешённом обслуживании поток A имеет эффективную скорость $\mu_A=0{,}5$ авт./с, а ожидаемое прибытие $\lambda_A=0{,}2$ авт./с. За 20 секунд приближённое изменение очереди: $\Delta q_A=(0{,}2-0{,}5)\cdot 20=-6$ автомобилей; q_A уменьшается с 18 до 12. Расчёт иллюстративный: реальные интервалы разгона, насыщения и потерь времени требуют калибровки.

7

Расчётный пример: перекрёсток с двумя фазами (продолжение)

Показатель	Фиксированный режим	Адаптивный режим	Способ проверки
Средняя задержка	Базовая линия	Сравниваемое значение	По траекториям/симуляции
95-й перцентиль очереди	Базовая линия	Сравниваемое значение	По интервалам наблюдения
Пропускная способность	Базовая линия	Сравниваемое значение	По пересечениям линий
Нарушения ограничений	Должно быть 0	Должно быть 0	Проверка логики контроллера

8 Прототипирование и программная реализация

Для лабораторного прототипа Raspberry Pi может получать поток камеры, выполнять инференс, вести счётчик машин и отправлять агрегированные измерения по MQTT. На сервере или локальном контроллере строится панель мониторинга. Переключение реальных светофорных сигналов допускается только через предусмотренный проектом защищённый интерфейс и с соблюдением применимых нормативных требований.

8.1. Псевдокод интеллектуального цикла

1. Получить кадр с временной меткой.
2. Выполнить детекцию транспорта и обновить треки.
3. Зафиксировать новые пересечения контрольных линий.
4. Оценить очереди, интенсивность и время ожидания.
5. Отфильтровать измерения; проверить качество данных.
6. Если камера недоступна или качество ниже порога — перейти в предусмотренный безопасный режим.

8 Прототипирование и программная реализация (продолжение)

7. Рассчитать веса разрешённых фаз и предложить длительность зелёного сигнала.

8. Проверить все ограничения безопасности в контроллере.

9. Передать допустимую команду, записать решение и показатели в журнал.

Практические испытания проводят сначала на архивном видео и цифровой модели перекрёстка. Для сравнения алгоритмов необходимо использовать одинаковые сценарии интенсивности, долей поворотов, пешеходных запросов и погодных условий. Результаты следует представлять с доверительными интервалами и анализом ошибок детекции.

9 Ограничения, надёжность и этические аспекты

Компьютерное зрение может ошибаться при плохом освещении, сильном дожде, перекрытии объектов и изменении положения камеры. Важны диагностика отказа видеопотока, контроль времени ответа, мониторинг дрейфа данных и безопасный резервный режим. При обработке дорожного видео необходимо минимизировать хранение идентифицирующих изображений и ограничивать доступ к данным. Нельзя считать распознавание лиц обязательным для подсчёта транспорта.

Метрики надёжности включают доступность канала, процент пропущенных кадров, задержку 95-го перцентиля, MAE подсчёта машин, долю неверных назначений треков и частоту перехода в резервный режим. Обязательна проверка поведения при потере сети, отказе камеры и перезагрузке edge-устройства.

12 Заключение

Интеллектуальное светофорное управление объединяет сенсоры, edge-вычисления, нейронные детекторы, трекинг, оценку транспортных потоков и алгоритмы принятия решений. Нейросетевая модель — лишь измерительная часть системы: качество управления зависит от достоверности данных, корректной динамической модели и соблюдения ограничений безопасности. Для научной оценки необходимо сравнение с фиксированным и классическим адаптивным управлением на одинаковых сценариях.

13 Контрольные вопросы

1. Чем адаптивное управление светофором отличается от фиксированного?
2. Какие задачи решает компьютерное зрение на перекрёстке?
3. Почему количество машин в кадре не равно интенсивности потока?
4. Какие функции выполняют edge-компьютер и промышленный контроллер?
5. Почему важно разделять обучающие данные по видеозаписям?
6. Что означают Precision, Recall, F1 и IoU?
7. Почему для подсчёта машин необходим трекинг?
8. Как вычисляется интенсивность потока за окно наблюдения?
9. Как оцениваются длина очереди и задержка автомобилей?
10. Каковы преимущества и ограничения сглаживания измерений?
11. Какие ограничения обязательны при переключении светофорных фаз?

13 Контрольные вопросы (продолжение)

12. Как предотвратить длительное ожидание малонагруженного направления?
13. Какие показатели следует сравнивать при эксперименте?
14. Как система должна действовать при отказе видеокамеры?
15. Какие вопросы приватности возникают при видеонаблюдении?

Рекомендуемая литература и ресурсы

1. Перри Ли. Архитектура интернета вещей. М.: ДМК Пресс, 2018. (Основная литература силлабуса.)

2. Arduino Documentation. https://docs.arduino.cc/learn/ (Ресурс силлабуса.)

3. MathWorks ThingSpeak Documentation. https://www.mathworks.com/help/thingspeak/ (Ресурс силлабуса.)

4. OpenCV Documentation. https://docs.opencv.org/ (Дополнительный ресурс.)

5. Ultralytics YOLO Documentation. https://docs.ultralytics.com/ (Дополнительный ресурс; версии и лицензии уточняются перед внедрением.)

6. FHWA Traffic Signal Timing Manual. Федеральное руководство США по принципам светофорного регулирования (дополнительный методический ресурс; не заменяет нормы Республики Казахстан).