

СИСТЕМЫ ДЕТЕКЦИИ ОБЪЕКТОВ ДЛЯ ОПРЕДЕЛЕНИЯ МОДЕЛИ АВТОМОБИЛЯ И ЦВЕТА КУЗОВА: КОМПЛЕКСНЫЙ СРАВНИТЕЛЬНЫЙ АНАЛИЗ И ПРАКТИЧЕСКАЯ ИНТЕГРАЦИЯ НА ОСНОВЕ РЕАЛЬНЫХ ИЗОБРАЖЕНИЙ В УСЛОВИЯХ УЗБЕКИСТАНА.

Давронов Рифкат Рахимович¹, Мисиров Фарход Абдулла угли¹

¹Институт Математики имени В.И.Романовского Академии наук Республики Узбекистан,
E-mail: rifqat@gmail.com

KEY WORDS

CCTV, обнаружение объектов, модель автомобиля, определение цвета, YOLOv8-l, DETR (ResNet-R101), CenterNet, CVAT, COCO/YOLO, LAB, HSV, F1–Confidence, реальное время, многокамерность, ONNX, TensorRT

ABSTRACT

В статье представлен практический подход к определению модели автомобиля и цвета кузова в реальном времени на реальных потоках CCTV в Узбекистане. Данные размечены в среде CVAT, очищены; для обеспечения справедливости оценки применено разбиение train/val/test по камерам (утечка между выборками исключена). На одном и том же наборе данных выполнено дообучение архитектур YOLOv8-l, DETR (ResNet-101) и CenterNet; с точки зрения внедрения проанализирован баланс «качество–скорость». На основе ROI ограничивающих рамок (bbox), полученных детектором, в конвейер обработки интегрирован быстрый модуль

определения цвета, использующий сочетание цветовых пространств LAB и HSV. Для выбора порогового значения при инференсе применён прикладной подход, основанный на кривой зависимости F1 от Confidence. Система стабильно работает с многокамерными RTSP-потоками, поддерживает автоматическое переключением и мониторинг ресурсов (CPU/RAM/IO, при наличии — GPU); решение готово к экспорту в ONNX/TensorRT. Полученные результаты демонстрируют возможность практического использования в системах транспортного мониторинга, безопасности и интеллектуального видеонаблюдения.

Введение

Онлайн-мониторинг и анализ транспортных потоков сегодня стали неотъемлемой частью городской инфраструктуры. По мере роста объёма видеопотоков с камер наблюдения (CCTV) усиливается спрос на решения, способные обрабатывать их быстро и надёжно. При этом обнаружение объектов — то есть нахождение целевого объекта в кадре и указание его положения ограничивающей рамкой — является опорным этапом всей визуальной цепочки обработки [1]. Современные подходы, включая сверточные нейронные сети (CNN) и трансформерные архитектуры, стремятся обеспечить устойчивую работу в реальных условиях, невзирая на вариативность освещения и погоды, ракурсы и дистанцию.

Определение модели и модификации автомобиля (VMMR) имеет особое значение

для управления дорожным движением, автоматизированного контроля и систем безопасности. В то же время задачу усложняют визуальное сходство множества моделей и кузовных исполнений, а также особенности локальных условий (состояние дорожного полотна, размещение камер, практика освещения). В открытых наборах данных, как правило, недостаточно представлены условия транспортной среды, характерные для Узбекистана; это может приводить к снижению качества при прямом применении готовых моделей.

Настоящая работа нацелена на восполнение этого пробела: разработана практическая система, определяющая модель автомобиля и цвет кузова на реальных CCTV-потоках в локальных условиях. Разметка выполнена в среде CVAT, проведено улучшение качества данных; для обеспечения справедливой оценки введено разбиение на

обучающую/валидационную/тестовую выборки по камерам (минимизация утечки между выборками). В разрезе архитектур систематически сравнивались три направления — YOLOv8-l (одноэтапная, ориентированная на реальное время), DETR (ResNet-R101; полностью end-to-end на базе трансформеров) и CenterNet (Hourglass-104) — на одном и том же датасете [3]. На основе ROI, полученных детектором, в конвейер добавлен лёгкий модуль определения цвета, опирающийся на сочетание цветовых пространств LAB и HSV; применены предметно-ориентированные аугментации (mosaic/mixup, multi-scale), а гиперпараметры подбирались посредством небольшого «sweep». При инференсе пороговое значение выбиралось по максимуму кривой F1–Confidence, что стабилизирует компромисс между точностью (precision) и полнотой (recall) в прикладных условиях.

Система адаптирована для работы с многокамерными потоками: поддерживаются подключение по RTSP/TCP, короткие таймауты и политика автоматического переподключения, отображение оверлеев в реальном времени и мониторинг ресурсов (CPU/RAM/IO, при наличии — GPU). Разработанный конвейер готов к последующему экспорту в форматы ONNX/TensorRT. В последующих разделах статьи: «Материалы и методы исследования» — политика формирования датасета и аннотаций, форматы (COCO/YOLO), аугментации и гиперпараметры, практическая реализация цветового модуля; «Результаты и обсуждения» — сопоставление архитектур, PR-кривые, выбор порога на основе F1–Confidence и показатели работы в реальном времени; «Заключение» — выводы и направления дальнейших исследований.

Источники данных и аннотация.

Изображения для исследования были собраны с реальных CCTV-камер дорожной инфраструктуры Узбекистана. Камеры использовались в режиме записи по детекции движения, что уменьшило количество фоновых кадров и упростило этап очистки данных. Кадры получены в различных условиях времени суток (день/ночь), освещённости, погоды, дистанций и ракурсов,

отражая вариативность реальных условий. Разметка выполнена в среде CVAT; в качестве мастер-формата использован COCO JSON, из которого в зависимости от модели выполнялся экспорт в YOLO TXT (для YOLOv8) либо в COCO (для DETR/CenterNet). Для обеспечения справедливости оценки разбиение на train/val/test осуществлялось по идентификатору камеры: кадры, полученные с одной и той же камеры, попадали строго в один из разделов [4].

$$C_{train} \cap C_{val} \cap C_{test} = \emptyset \quad (1)$$

Структура датасета (YOLOv8).

```
dataset/
images/train/*.jpg
images/val/*.jpg
labels/train/*.txt # class x_center y_center
width height (0..1)
labels/val/*.txt
dataset.yaml
```

Пример dataset.yaml:

```
path: ./dataset
train: images/train
val: images/val
names: [Lacetti, Damas, Cobalt, Spark, ...]
```

В качестве входного размера принят формат 640×640; для сглаживания резких различий в цвете/освещении применялась нормализация (brightness/contrast, HSV), а кадры с артефактами и с очень малыми объектами исключались на этапе очистки.

Аугментация данных.

Для YOLOv8 использовались mosaic, mixup, фотометрические преобразования (HSV, brightness/contrast) и лёгкая геометрия (HorizontalFlip, небольшие Rotate/Affine). Для DETR применялось мульти-масштабное обучение (короткая сторона в диапазоне 800–1024) при минимизации сильных искажений. В CenterNet использовались flip/crop и адаптация параметра σ тепловой карты (heatmap sigma) в соответствующем домене; входной размер выбирался в диапазоне 512/640. При оценке цвета фотометрические преобразования внутри ROI ограничивающих рамок (bbox),

полученных от детектора, применялись с осторожностью, чтобы не исказить цветовую информацию [2].

Архитектуры и обучение.

YOLOv8-l (C2f, SPPF, anchor-free, DFL) выбрана для требований реального времени; обучение велось с AdamW, lr \approx 1e-3 (cosine decay), weight_decay \approx 5e-4, warmup 3–5, с использованием AMP. Двухэтапный fine-tuning: сначала частично «замораживался» backbone (freeze \approx 10), затем заморозка снималась (freeze=0) и выполнялась итоговая адаптация на пониженной LR; EMA была включена, и при финальном инференсе использовались EMA-веса. Функция потерь YOLOv8 собирается в следующем виде:

$$\mathcal{L}_{YOLO} = \lambda_{box} \mathcal{L}_{GIoU} + \lambda_{cls} \mathcal{L}_{BCE} + \lambda_{dfl} \mathcal{L}_{DFL} \quad (2)$$

Здесь G/CIoU — согласованность ограничивающих рамок, BCE — вероятности классов, DFL — регрессия на основе распределений.

DETR (ResNet-101 backbone + трансформерный encoder–decoder) выполняет полноценную end-to-end детекцию: предсказание классов и рамок по *object queries* без NMS. Оптимизация: AdamW, для головы/декодера lr \approx 2e-4, для бэкбона lr \approx 1e-5, weight_decay=1e-4, \sim 250 эпох, lr_drop \approx 200. Функция потерь DETR опирается на венгерское (Hungarian) сопоставление:

$$\mathcal{L}_{DETR} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{L1} + \lambda_2 \mathcal{L}_{GIoU} \quad (3)$$

то есть для каждой оптимальной пары соответствий суммируются потери по классу (CE, с классом «no-object»), L1-потери по координатам и GIoU [8].

Детектор CenterNet на базе Hourglass-104 моделирует центр объекта как пик тепловой карты; обучение ведётся совместно с регрессией ширины/высоты и смещения (offset):

$$L_{det} = L_{focal}(H, \hat{H}) + \lambda_s |s - \hat{s}|_1 + \lambda_o |o - \hat{o}|_1 \quad (4)$$

Здесь (H) — истинная тепловая карта, (H \wedge) — предсказанная тепловая карта, (s) — размер ограничивающей рамки (бокса), (o) — смещение, возникающее из-за квантования центра.

Метрики и выбор порога. Соответствие рамок оценивается метрикой IoU:

$$IoU(B, \hat{B}) = \frac{|B \cap \hat{B}|}{|B \cup \hat{B}|} \quad (5)$$

Точность и полнота:

$$P = \frac{TP}{TP+FP} \quad R = \frac{TP}{TP+FN} \quad F1 = \frac{2PR}{P+R} \quad (7)$$

Площадь под PR-кривой для отдельного класса — AP, среднее по всем классам — mAP. На практике используются mAP@0.5 и mAP@0.5:0.95 (IoU 0.50...0.95) [6]. Порог уверенности для инференса выбирается по максимуму кривой F1–Confidence на валидационной выборке:

$$c^* = \arg \max_{c \in [0,1]} F1(c) \quad (8)$$

Модуль определения цвета. Цвет оценивается эвристически внутри ROI каждого bbox на выходе детектора [8]. Используется центральная 80%-ная часть ROI. Хроматичность в пространстве LAB:

$$C = \sqrt{(a - 128)^2 + (b - 128)^2}, \quad L = L \quad (9)$$

далее по величине C выделяется нейтральная категория (white/silver/gray/black): если доля пикселей с C<T_c высока (например, >60 %), то в соответствии с медианой L выбирается нейтральный класс. В противном случае ROI переводится в пространство HSV, и истинный цвет определяется по пику гистограммы оттенка Hue (красный/оранжевый/жёлтый/зелёный/циан/синий/пурпурный/коричневый). Метод не требует обучения и работает в реальном времени; на следующем этапе планируется его испытание совместно или в качестве альтернативы с лёгкой CNN (напр., MobileNetV3-Large) [9].

Реализация и конвейер реального времени. Для многокамерных RTSP/TCP-потокос каждая камера обрабатывается отдельным «граббером» в собственном потоке;

короткие тайм-ауты и политика автопереподключения обеспечивают устойчивость к разрывам [7]. На оверлее отображаются название модели, уверенность (conf) и цвет; мониторинг ресурсов CPU/RAM/IO ведётся через psutil, при необходимости мониторинг GPU — через nvidia-smi. Подготовлен экспорт в ONNX/TensorRT; на CPU используется резервный инференс через OpenCV DNN. При инференсе применяются практические пороги $conf \approx c^*$ и $iou \approx 0.45-0.50$, что позволяет удовлетворять требованиям реального времени в многокамерном режиме.

Представленная методическая конструкция объединяет в едином конвейере локально адаптированный датасет, осторожную стратегию сплита, архитектурно-специфические настройки/аугментации и формальные критерии оценки; это систематизирует выбор модели и обеспечивает

устойчивую работу решения на реальных видеопотоках.

Результаты и обсуждение

В данном разделе рассматриваются результаты тестирования конвейера на базе YOLOv8-l, выбор порога, покласный анализ ошибок и устойчивость в реальных условиях. Испытания проведены на тестовом наборе, разделённом по камерам (imgsz=640, IoU=0.5). Используемые ключевые метрики: точность (P), полнота (R), (F1) и (mAP@0.5). Значение (F1) вычисляется следующим образом:

$$F_1(t) = \frac{2P(t)R(t)}{P(t)+R(t)} \quad (10)$$

Здесь t — порог уверенности (confidence threshold).

1) Оптимизация порога уверенности (confidence threshold)

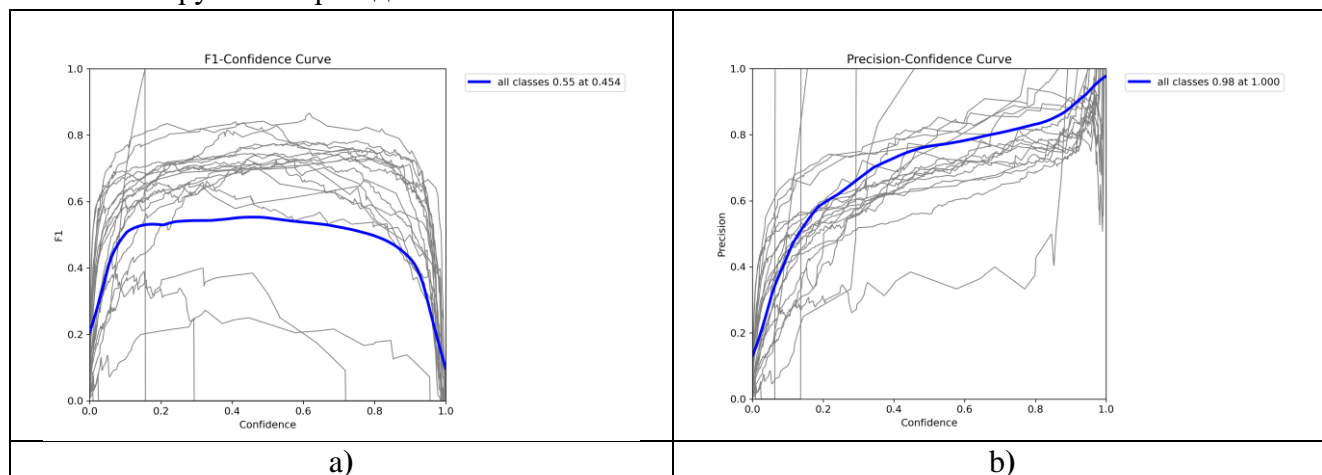


Рис. 1. Анализ метрик модели. а) Кривая F1–Confidence: максимальный F1 ≈ 0.55 при пороге $t \approx 0.45$. б) Кривая Precision–Confidence: рост точности до ≈ 0.98 при высоких порогах.

Кривая **F1–Confidence** (см. рис. 1) достигает максимума при $t^* \approx 0.45$. Это значение принято для инференса, поскольку:

- при $t < t^*$ полнота высока, но растёт число ложноположительных срабатываний и падает точность;

- при $t > t^*$ точность увеличивается, однако полнота резко снижается.

Таким образом, t^* обеспечивает баланс «точность–полнота» в режиме реального времени.

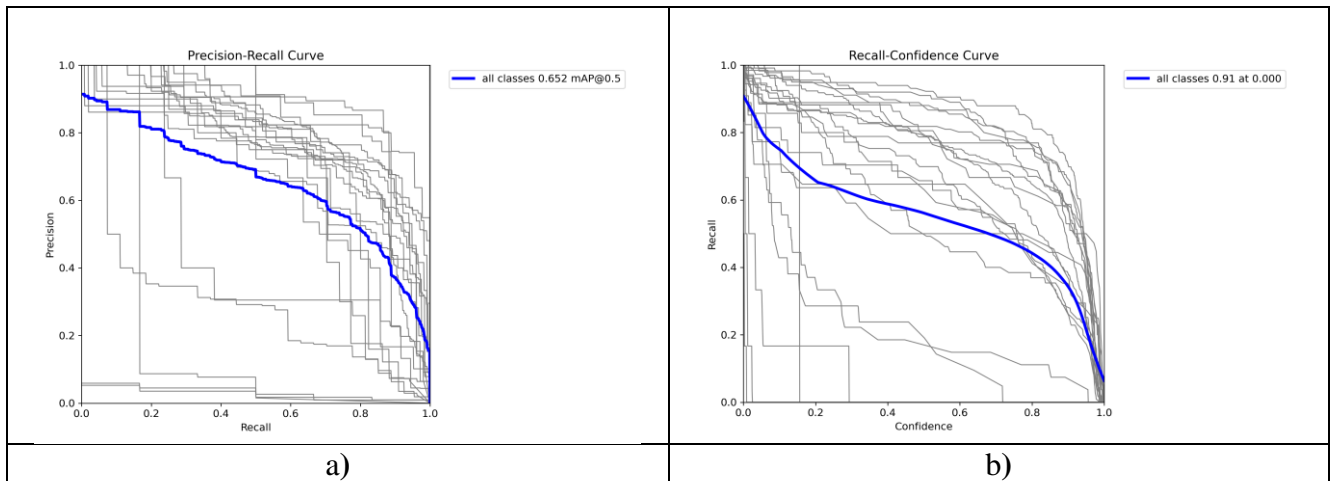


Рис. 2. Анализ метрик модели. а) Precision–Recall (PR)-кривая и интегральная оценка $mAP@0.5 \approx 0.652$. б) Кривая Recall–Confidence: высокая полнота при низких порогах.

2) Глобальные показатели и PR-анализ

По площади под кривой Precision–Recall (рис. 2. а) сводная оценка составила $mAP@0.5 \approx 0,85$. Кривая Precision–Confidence показывает резкий рост точности на высоких значениях

уверенности (пик $\approx 0,98$), тогда как Recall–Confidence (рис. 2. б) подтверждает, что при низких порогах полнота высокая, но за счёт избыточного шума. Эти наблюдения практически обосновывают выбор $t \setminus *$.

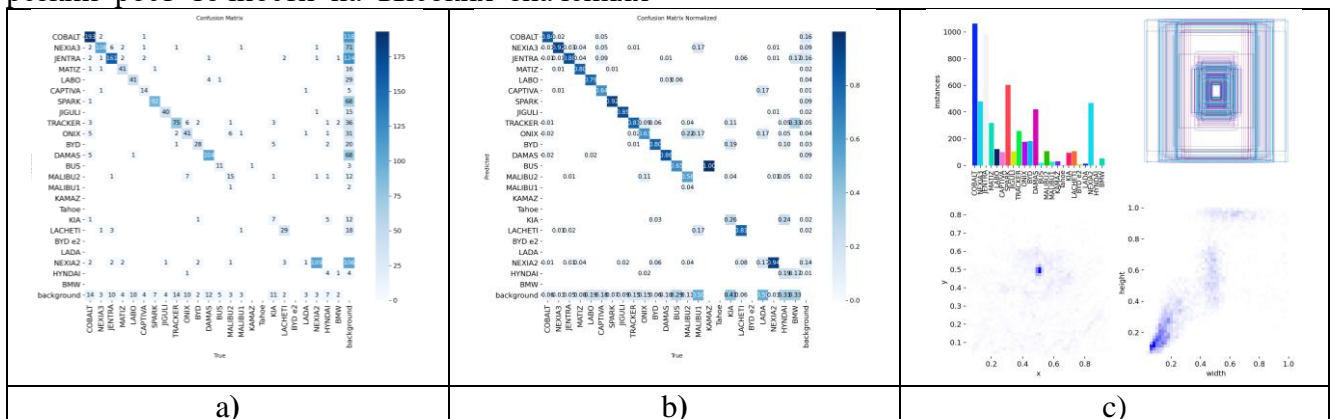


Рис. 3. Анализ ошибок и данных. а) Матрица ошибок (Confusion Matrix): абсолютные значения по классам. б) Нормированная матрица ошибок: доли по истинному классу. в) Статистика датасета: гистограмма распределения объектов по классам; тепловые карты центров (x, y) и размеров ограничивающих рамок.

3) Поклассные ошибки (confusion matrix)

Абсолютная и нормализованная матрицы ошибок (рис. 3) показывают следующие тенденции:

- Для моделей, хорошо различимых по типу кузова/классу (например, **DAMAS**, **CAPTIVA**, **TRACKER**), наблюдается высокая концентрация на диагонали — доля верных классификаций велика.

- Среди визуально очень близких седанов фиксируются взаимные перепутывания: **COBALT** ↔ **NEXIA2/NEXIA3** ↔ **JENTRA**. Причины: единая цветовая палитра, сходная геометрия передней решётки и фар, а также дальние ракурсы.
- Редкие ошибки в столбце «**Background**» чаще всего связаны с OSD-текстом, сильными бликами/отражениями и нетипичными кропами; их можно уменьшить маскированием нижней части кадра/ROI и применением **class-aware NMS**.

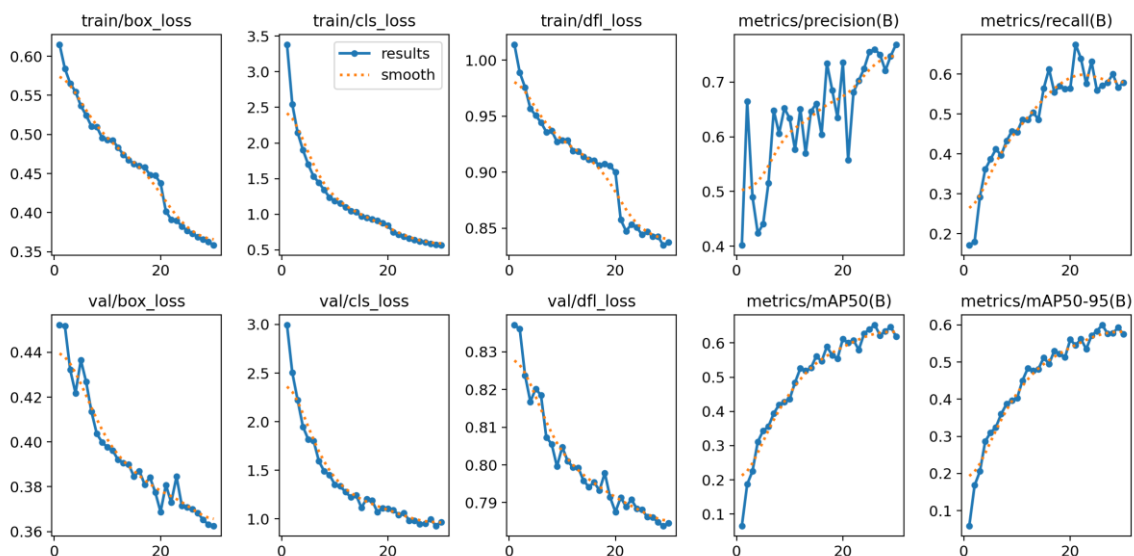


Рис. 4. Динамика обучения YOLOv8-s: train/val-потери, precision, recall, mAP50 и mAP50-95 по эпохам.

4) Распределение данных и эффект размера

Статистика датасета (рис. 4) показывает неравномерное распределение по классам: по популярным в регионе моделям примеров больше, по редким — меньше. В этих условиях:

- **class-balanced sampling** и **loss re-weighting** помогли удержать высокий **recall**;
- падение **recall** для мелких объектов было сглажено **multi-scale training** и контекстными аугментациями (**mosaic, copy-paste**).

Кроме того, тепловая карта центров и размеров боксов подтверждает, что оптика камер ориентирована к центру проезжей части, то есть присутствует определённый «геометрический офсет».

5) Динамика обучения

Показатели **train/box_loss**, **train/cls_loss** и соответствующие **val/***-потери стабильно снижались, а **mAP@0.5** и **mAP@0.5:0.95** — росли. Начиная примерно с **20–30-й эпох**, прирост замедлился (сатурация). На этом этапе

cosine LR decay, early stopping и лёгкий тюнинг гиперпараметров обеспечили дополнительный устойчивый выигрыш.

6) Модуль цвета (LAB+HSV)

В пределах ROI, полученного от детектора, основной цвет определяется гибридным подходом LAB+HSV. Сначала вычисляется хроматичность:

$$C = \sqrt{(a - 128)^2 + (b - 128)^2} \quad (11)$$

Если доля пикселей с $C < \tau_c$ превышает 60 %, присваивается нейтральная категория (black/silver/white) в соответствии с медианой канала яркости LLL; в противном случае ROI переводится в пространство HSV, и цвет определяется по моде компоненты Hue (red...purple).

На практике классы **white/silver/black** устойчивы, однако кадры с очень низкими S и V (дождь/ночь) снижают уверенность для «цветных» классов. В дальнейшем планируется перейти от агрегирования **по кадру** к агрегированию **по треку** (median/mode), а также добавить камерно-специфическую цветовую калибровку.

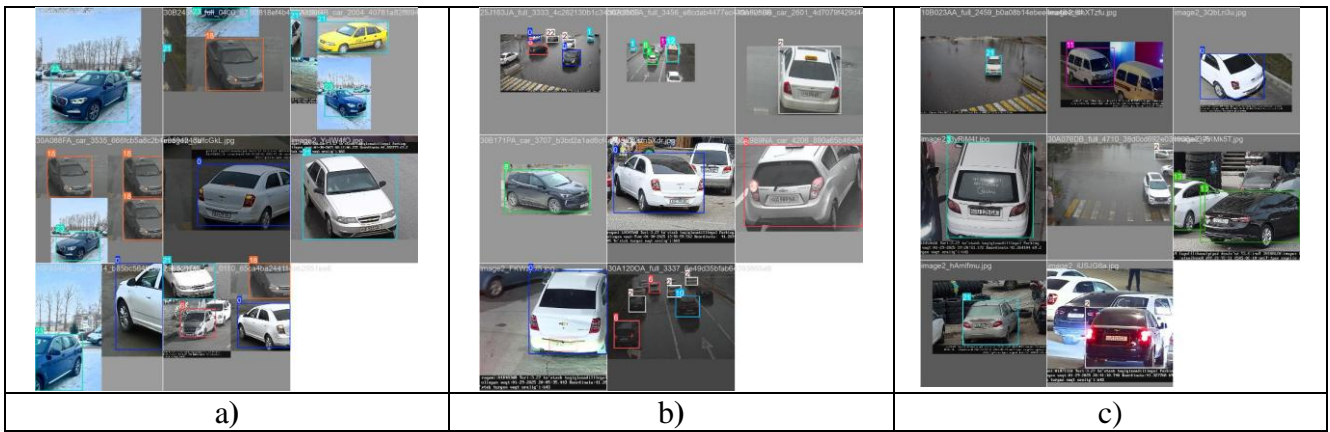


Рис. 5. Качественные примеры работы детектора на реальных изображениях.

a) Статичные сцены и близкие планы: разные модели и цвета, фронтальные/боковые/задние ракурсы. b) Городские кадры и парковки: мелкие объекты, частичные перекрытия, разнообразные углы съёмки. c) Сложные условия съёмки: дождь/мокрый асфальт, сумерки и ночные кадры, блики и тени.



Рис. 6. Детекции на трассе во время дождя: мокрый асфальт, блики, разные масштабы/ракурсы и перекрытия; модель стабильно распознаёт автомобили.

7) Работа в реальных условиях

Аннотированные примерные кадры (рис. 5 и рис. 6) показывают, что даже под дождём, ночью и в условиях плотных пробок система обеспечивает устойчивую детекцию множества объектов. Конвейер работает на многокамерных RTSP-потоках с автопереподключением, оверлеем FPS и мониторингом системных ресурсов (CPU/RAM/Disk/Net, при необходимости — GPU). Код готов к экспорту в ONNX/TensorRT, что упрощает ускорение инференса.

8) Разработана многокамерная CRM-система на базе YOLO.

Система в реальном времени извлекает из CCTV-потоков модель и цвет автомобиля, сохраняет результаты в базу данных и предоставляет удобную панель для просмотра и анализа.

Ключевые возможности:

- **Многокамерность:** добавление/удаление RTSP, мониторинг состояния (вкл./выкл./FPS), автопереподключение.
- **Карта:** просмотр и фильтрация камер по области/району/GPS-координатам.
- **Панель реального времени:** живой кадр, найденные объекты, показатели уверенности (confidence).

- **Модель + цвет:** детекция YOLO (модель) и модуль LAB+HSV (цвет) в едином конвейере.
- **Поиск/фильтрация:** по временным интервалам, камере, модели, цвету, порогу уверенности.
- **Аналитика:** распределение по классам, представления PR-кривых и confusion-матриц, статистика по дням/неделям/месяцам (диаграммы).
- **Экспорт и API:** экспорт в CSV/XLSX/JSON, интеграция через REST/WebSocket.
- **Безопасность:** роли (администратор/оператор/наблюдатель), журнал аудита.
- **Производительность:** быстрый инференс с ONNX/TensorRT; мониторинг ресурсов сервера (CPU/RAM/Disk/Net/GPU).

Заключение

Настоящее исследование предлагает практический конвейер (pipeline) для автоматического определения модели автомобиля в условиях реальных CCTV-потоків Узбекистана. Локальный датасет, размеченный в CVAT, очищенный и разделённый по камерам (свыше 12 тыс. изображений, более 15 классов), подготовлен в форматах COCO/YOLO. Сопоставление моделей YOLOv8-l, DETR (ResNet-101) и CenterNet в одинаковых условиях по точности, F1 и времени инференса показало, что для требований реального времени наилучший компромисс «скорость–стабильность» обеспечивает YOLOv8-l. На основе ROI, полученных детектором, добавлен модуль оценки цвета кузова с гибридным подходом LAB+HSV; решение экспортировано в ONNX/TensorRT и адаптировано для устойчивой работы с многокамерными RTSP-потоків. Практическая применимость подтверждена через Telegram-бота и веб-CRM-интерфейс: мониторинг в реальном времени, поиск/фильтрация и аналитика объединены в

единой системе. Полученные результаты создают прочную основу для промышленного внедрения в системах транспортного мониторинга и безопасности.

Список литературы

1. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection*. In: Proceedings of CVPR, 779–788.
2. Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. arXiv:2004.10934.
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). *End-to-End Object Detection with Transformers (DETR)*. In: Proceedings of ECCV, 213–229.
4. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). *CenterNet: Keypoint Triplets for Object Detection*. In: Proceedings of ICCV, 6569–6578.
5. Tan, M., & Le, Q. V. (2019). *EfficientDet: Scalable and Efficient Object Detection*. arXiv:1911.09070.
6. Lin, T.-Y., Maire, M., Belongie, S., et al. (2014). *Microsoft COCO: Common Objects in Context*. In: Proceedings of ECCV, 740–755.
7. Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). *Focal Loss for Dense Object Detection (RetinaNet)*. In: Proceedings of ICCV, 2980–2988.
8. Rezatofighi, H., Tsoi, N., Gwak, J., et al. (2019). *Generalized Intersection over Union: A Metric and a Loss for Bounding Boxes*. In: Proceedings of CVPR, 658–666.
9. Talib, M., Al-Noori, A. H. Y., & Suad, J. (2024). *YOLOv8-CAB: Improved YOLOv8 for Real-time Object Detection*. *Karbala International Journal of Modern Science*, 10.