

Автоматизация и конвейер CV-моделей для авиационного обслуживания

Введение

В гражданской авиации (в частности, в техническом обслуживании и ремонте, ТОиР, и наземном обслуживании) растёт интерес к применению компьютерного зрения для автоматизации контроля и безопасности. Задачи включают проверку соблюдения регламентов обслуживания (например, подключение оборудования в нужное время), мониторинг подъезда спецтранспорта, буксировки самолётов, подключения обогревателей и прочих операций. Для решения этих задач требуется конвейер моделей CV (computer vision) от сбора и разметки данных до обучения моделей и их деплоя. В таком конвейере могут использоваться самые разные типы алгоритмов — от простой классификации изображений до детекции объектов, сегментации областей и оценки позы людей. Ниже рассмотрены актуальные подходы и инструменты, включая новейшие методы разметки (Segment Anything Model 3), современные архитектуры детектирования (гибриды свёрточных сетей и трансформеров, на примере моделей YOLO и RF-DETR), а также платформы вроде Roboflow для максимально **безкодowego** построения решений.

Новейшие подходы к разметке данных (SAM и др.)

Трудоёмкая разметка изображений и видео — узкое место при создании CV-моделей. Современный тренд — использовать мощные модели сегментации для ускорения и упрощения разметки. Наиболее заметный пример — **Segment Anything Model 3 (SAM 3)** от Meta. SAM 3 — это модель сегментации общего назначения нового поколения, которая способна **по текстовому или визуальному запросу обнаруживать, сегментировать и даже отслеживать объекты на изображениях и видео** ¹ ². В отличие от ранних версий, SAM 3 поддерживает **открытую словарную сегментацию**: например, можно задать текстовый промпт «жёлтый школьный автобус», и модель выделит масками все объекты, соответствующие этому описанию ³. Это превращает SAM из инструмента геометрической сегментации в модель концептуального уровня, способную находить сразу все экземпляры запрошенного класса на изображении ³ ⁴. Кроме того, SAM 3 может работать и по одному визуальному примеру (показываем изображение объекта) или классическим способом (точка/прямоугольник на объекте), и поддерживает **трёхмерные сценарии** (связанная модель SAM 3D способна по одному 2D-снимку реконструировать 3D-объекты или оценивать позу тела) ¹ ⁵.

Пример использования Segment Anything Model 3: модель по текстовому запросу находит и сегментирует все указанные объекты на изображении (в данном случае — «shipping container», контейнеры отмечены цветными масками). SAM 3 выполняет Promptable Concept Segmentation — сегментацию всех объектов указанного концепта ⁶ ⁴.

Производительность и требования SAM 3: Несмотря на внушительный размер (~840 млн параметров, вес модели ≈3,4 ГБ), SAM 3 оптимизирован по памяти и скорости. Она потребляет **меньше видеопамати на инференсе, чем SAM 2, и свободно помещается на GPU с 16 ГБ VRAM** ⁷ ⁸. Meta указывает, что на серверном GPU Nvidia H200 (последнего поколения) модель обрабатывает изображение (~100 объектов) примерно за 30 мс ⁷. Это означает, что на более

доступном GPU вроде **RTX 4080 с 16 ГБ** можно достичь приемлемой скорости работы для нестрогих real-time задач (пусть throughput будет ниже, но модель **реально запустить на 4080 без ухищрений** и даже с небольшим батчем изображений) ⁹. В реальных сценариях разметки и офлайн-анализа (например, интерактивная сегментация через веб-интерфейс, разметка датасетов, периодическая обработка кадров с камер) связка SAM 3 + RTX 4080 показывает себя более чем достаточной. При этом SAM 3 — **серверного класса модель**, и для прямого применения на краю (edge) или строгого real-time её тяжесть может быть ограничением ¹⁰. Однако, важно отметить, что SAM 3 предоставляет отличный способ ускорить подготовку данных: **её можно использовать для автоматической разметки**. Meta и исследователи рекомендуют такой workflow: сначала применить SAM 3 для разметки и поиска нужных объектов на изображениях, а затем на основе размеченного датасета обучить более лёгкую целевую модель детекции ¹⁰. Такой маленький детектор уже не будет обладать текстовыми “open-vocabulary” возможностями SAM, зато сможет в реальном времени находить именно те объекты, которые были обучены, гораздо быстрее ¹⁰. В наших задачах это значит, что можно, например, с помощью SAM быстро размечать снимки самолётов и обслуживающей техники (сам аэродромный тягач, обогреватели, шланги, персонал и т.д.), а потом обучить компактную модель, которая будет **оперативно мониторить видео-поток** на предмет этих объектов.

Помимо SAM, существуют и другие методы AI-ассистированной разметки. Например, модели вроде **Grounding DINO** способны по текстовому запросу обвести объекты прямоугольниками, а **другие версии Segment Anything (SAM 1, SAM 2)** или аналогичные сегментаторы могут применяться для различных доменов (медицинские снимки, спутниковые кадры и т.п.). Однако SAM 3 на конец 2025 г. является одним из самых передовых и универсальных инструментов: он продемонстрировал высокую надёжность на редких и невиданных объектах, умеет точно сегментировать мелкие и слабо контрастные детали, отделять соприкасающиеся объекты с чёткими границами ¹¹ ¹². В целом, новейшие foundation-модели сегментации вроде SAM 3 позволяют значительно ускорить создание разметки **без ручного рисования каждого контура**, что критично для больших индустриальных датасетов.

Обнаружение объектов: гибридные архитектуры (YOLO и DETR)

Для автоматического обнаружения и отслеживания ключевых объектов (спецтехника, персонал, элементы самолёта и пр.) в потоковом видео традиционно используются модели детекции объектов. В последние годы архитектуры детекторов эволюционировали от чисто свёрточных к гибридным, включающим механизмы трансформеров. Классический представитель одностадийных детекторов – **YOLO (You Only Look Once)**, семейство моделей, известных своей высокой скоростью, — к 2025 г. дорос до версии YOLOv12 ¹³. YOLO обрабатывает всё изображение за один прогон сети (one-stage), предсказывая bounding box’ы и классы напрямую, без отдельного этапа region proposal, что даёт ему преимущество в **real-time производительности** ¹⁴. Каждый новый выпуск YOLO приносил улучшения точности и эффективности; например, современные версии существенно повысили качество по сравнению с YOLOv3/v4, сохранив простоту развёртывания. В то же время, исследователи начали включать в YOLO-прототипы и элементы трансформеров: уже упоминаются гибридные модели **YOLO-S**, объединяющая свёрточный экстрактор признаков с self-attention для быстрого инференса, и **YOLOv12**, в которой используются рекуррентные или последовательные блоки (sequence learning) наряду с трансформерными механизмами ¹⁵. Такие гибриды пытаются взять лучшее от обоих подходов – локальную эффективность ConvNet и долгосрочные зависимости Vision Transformer’ов.

Параллельно с эволюцией YOLO, Facebook AI Research предложила совершенно иную архитектуру детекторов – на базе **Detection Transformer (DETR)**. DETR и его последователи используют трансформер для прямого предсказания набора объектов на изображении, **избегая NMS** (процедуры отбора пересекающихся предсказаний) и других эвристик. Ранние DETR требовали очень долгого обучения и были медленнее на inference, но к 2023–2025 гг. появились улучшения (Deformable DETR, DINO, etc.), значительно ускорившие сход и повысившие точность. Более того, оказалось, что **с учётом времени на NMS у YOLO**, современные DETR могут почти сравняться по задержке выдачи результата ¹⁶. То есть если считать **полное время до получения готовых боксов**, трансформерные детекторы не уступают YOLO, а по точности часто выигрывают, особенно при наличии хорошего pre-training ¹⁶ ¹⁷. Предварительное обучение на больших данных даёт DETR-моделям сильную обобщающую способность, что важно в наших узкоспециализированных задачах: модель с **мощным бекбоном и языковым пониманием** способна быстрее адаптироваться к небольшим кастомным датасетам ¹⁸.

Особое внимание в сообществе привлекла модель **RF-DETR** – Real-Time Detection Transformer, разработанный компанией Roboflow. Это **облегчённый DETR, спроектированный автоматически (NAS)** под требования реального времени ¹⁹ ²⁰. RF-DETR стал первым трансформерным детектором, который **превысил порог 60 mAP на COCO при работе в реальном времени** ²¹. Причём достигнуто это на относительно компактных моделях: заявлены два варианта – базовый ~29 млн параметров и большой ~128 млн параметров ²². Для сравнения, это сопоставимо или даже меньше, чем у многих версий YOLO. Тем не менее, RF-DETR показывает **SOTA-качество на реальных данных** и универсальность к разным доменам, оставаясь достаточно быстрым для работы на GPU среднего класса или даже на некоторых edge-устройствах ²⁰. В бенчмарках Roboflow отмечается, что **RF-DETR превосходит или соответствует** лучшим существующим моделям как среди трансформерных (например, D-FINE, LW-DETR), так и среди свёрточных (YOLOv8, гипотетический YOLO11) по совокупности метрик точности и скорости ²³ ²⁴. Таким образом, в практике 2025 года при построении системы обнаружения объектов можно рассматривать два подхода: продолжать использовать отточенные временем **YOLO (особенно если нужен максимально простой и лёгкий вариант, в том числе для edge)**, либо переходить на **новые DETR-подобные модели** (как RF-DETR) с открытой лицензией Apache 2.0 ²¹, которые обещают более высокую точность и лучшую переносимость на разные датасеты. Впрочем, реальный выбор может зависеть от имеющихся ресурсов и требований: например, если уже есть готовая модель YOLO под нужные классы – её легко интегрировать, но если задача требует **открытого набора классов** или тесной связи с текстовыми описаниями, то сочетание DETR-моделей с языковыми embedding'ами (CLIP, SAM3 и т.п.) может дать выигрыши.

Пример детекции объектов моделью YOLO: на изображении автоматически выделены рамкой енок и мяч, с указанием классов и уверенностей. Одностадийные модели YOLO славятся способностью работать в реальном времени, выполняя и распознавание, и локализацию объектов за один проход сети ¹⁴. К 2025 г. семейство эволюционировало до версии YOLOv12, постепенно повышая точность и удобство использования ¹³.

Стоит отметить, что в многих сценариях эффективна комбинация подходов: **гибридные системы**, где, например, лёгкий CNN-детектор работает постоянно, а тяжёлая модель вроде SAM подключается эпизодически. Так, упомянутый выше workflow (YOLO + SAM3) может решать задачу сегментации: YOLO быстро определяет объекты (скажем, нашёл спецтранспорт около самолёта), после чего SAM3 по запросу сегментирует точную область (контуры транспортного средства, точное положение агрегата обогрева на борту и т.д.). Это уменьшает общую нагрузку, поскольку тяжёлый сегментатор вызывается только по необходимости. Гибкость современных CV-инструментов позволяет также строить конвейеры «детекция -> трекинг -> действие» или

добавлять проверки на основе классификации состояния. Например, сперва детектор отмечает, что отопительная установка подключена (по наличию соединительного рукава), затем классификатор по изображению панели может определить, включена она или нет (по цвету индикатора), а модель позы людей может проверить, находится ли технический персонал в безопасной зоне при запуске агрегата.

Классификация и оценка позы в задачах наблюдения

Помимо обнаружения конкретных объектов, для ряда задач востребованы смежные методы CV: **классификация состояний/сцен** и **детекция ключевых точек (поза)**. Классификация на уровне всего изображения может применяться, например, чтобы определить, соответствует ли кадр определённому этапу технологического процесса. В качестве иллюстрации: можно обучить модель, которая по фото определяет, завершён ли предполётный обогрев двигателя (по наличию инея или по положению заслонок). Такие классификаторы проще обучаются (им нужен лишь метка на кадр в целом) и могут служить дополнительным слоем проверки. Современные подходы предлагают либо использовать свёрточные нейросети, либо более новые vision transformer для классификации; на практике же часто применяются предобученные модели (ResNet, EfficientNet, ViT и др.), дообученные на ваших данных. Например, сервисы AutoML могут без кода обучить классификатор, если вы загрузите положительные/отрицательные примеры кадров.

Другая категория — **оценка позы (pose estimation)**, то есть **выделение ключевых точек** на человеке или объекте для понимания его положения. В контексте обслуживания аэропортов это может быть полезно для мониторинга персонала: распознавание жестов сигналистов, контроль правильной осанки при подъёме тяжестей, распознавание, что техник встал в опасную зону (например, под поднятым крылом) по характерной позе. Pose estimation традиционно ассоциируется с человеческим скелетом (набор точек суставов), но может использоваться и для техники – например, отследить положение подвижных элементов (угол поднятия закрылков, положение шланга). Современные модели позы значительно продвинулись: имеются как легковесные, работающие на мобильных устройствах (например, **MediaPipe Pose** от Google, выдающая 33 ключевые точки тела в реальном времени на смартфоне), так и интегрированные в семейство YOLO – например, **YOLOv8-Pose** и новые **YOLO11 Pose** предлагают сочетание быстрого детектирования человека и предсказания его ключевых точек ²⁵. Сочетание CNN и transformer также улучшило точность позных моделей, оставаясь достаточными для edge-устройств ²⁵.

В экосистеме Roboflow появилось удобное средство работать с позой без кода: с 2024 г. платформа поддерживает задачи keypoint detection. Пользователь может **задать скелет из нужных точек** (например, для человека – стандартный набор суставов, либо кастомный набор для объекта), разметить изображения через интерфейс, и затем обучить кастомную модель прямо в облаке ²⁶ ²⁷. После обучения модель позы тоже разворачивается через API или на устройстве. Таким образом, даже сложные случаи, вроде определения позы сотрудника на видео с нагрудной камеры, можно решить, не погружаясь в детали кода – достаточно собрать небольшой датасет с правильной разметкой.

Пример задачи оценки позы человека: модель позной детекции определяет координаты ключевых точек (суставов) на теле и соединяет их в скелет. Современные алгоритмы (например, YOLOv8/11 Pose, MediaPipe) позволяют одновременно обрабатывать позы нескольких людей в кадре в режиме реального времени ²⁵. В платформе Roboflow поддерживается полный цикл работы с подобными моделями – от разметки скелетов до развёртывания готового детектора ключевых точек ²⁶.

Отметим, что объединение результатов разных моделей – распространённый приём. Например, в задаче фиксации нарушения правил можно **скомбинировать детекцию объектов и поз:** обнаружить человека и технику поблизости и параллельно определить позу человека, чтобы понять, смотрит ли он на приближающийся транспорт, жестом ли регулирует движение, или отвлёкся. Аналогично, оценка позы может применяться для отслеживания положения буксирного штыря относительно самолёта (через определение ключевых точек на механизме). Гибкость CV-пайплайна позволяет добавлять такие модули по мере необходимости.

Инструменты и платформы для безкодowego конвейера (Roboflow и аналоги)

Чтобы реализовать всё вышеперечисленное на практике, существуют **специализированные платформы, ускоряющие разработку CV-решений без (или с минимумом) программирования.** Одним из лидеров является **Roboflow** – комплексная end-to-end платформа компьютерного зрения. Roboflow предоставляет:

- **Управление данными и разметку:** удобный веб-интерфейс для загрузки изображений и видео (кадры извлекаются автоматически), разметки объектов прямоугольниками, полигонами или ключевыми точками. Поддерживается совместная работа, версии датасетов, конвертация форматов аннотаций. Есть и AI-ассистированные методы разметки – например, интеграция с моделями (в т.ч. SAM для авто-сегментации) и функциональность **Autodistill** (open-source инструмент для авторазметки с помощью предобученных моделей). Это позволяет значительно ускорить подготовку тренировочных данных.
- **Обучение моделей:** платформа предлагает облачную тренировку популярных моделей **в несколько кликов.** Поддерживаются как свёрточные детекторы (YOLOv5, YOLOv8, YOLOv7 и пр.), так и новые модели (например, упомянутый RF-DETR можно обучить через Roboflow Train)²⁸. Настройки (гиперпараметры, augmentation) по умолчанию оптимальны, так что даже новичок может получить рабочую модель. Roboflow также предоставляет свои оптимизированные бэкбоны (например, собственный начальный вес RF-DETR, дающий более высокий mAP, чем стандартный COCO-чекпойнт²⁹). В случае задачи классификации или сегментации – схожий процесс: выбирается тип модели, запускается обучение; необходимый GPU ресурсы платформа берёт на себя.
- **Деплоймент и интеграция:** после обучения модель можно **развернуть без инфраструктурных хлопот.** Roboflow предлагает хостинг модели с REST API (т. н. Infer API), с помощью которого можно отправлять запросы (из любого приложения, будь то веб-сервис или скрипт на борту самолёта) и получать результаты распознавания. Либо можно экспортировать веса модели в популярном формате (Torch, TensorRT, ONNX и т.п.) и разворачивать на своём оборудовании. Документация включает гайды по развёртыванию на **разнообразных устройствах** – от облачных серверов до **NVIDIA Jetson, Raspberry Pi, мобильных приложений**³⁰. Таким образом, возможен вариант вообще без написания кода: например, собранный через Roboflow детектор можно сразу использовать через готовый веб-вызов; для интеграции в промышленную систему достаточно написать логику обработки результатов (что также можно сделать графически, если использовать no-code workflow менеджеры). К слову, в Roboflow недавно появился модуль **Workflows**, предоставляющий low-code интерфейс для построения целых CV-приложений из блоков (каскадов моделей, триггеров, действий).

Не менее важен вопрос лицензирования и стоимости. **Условия использования Roboflow:** платформа имеет бесплатный базовый план (Public plan), которым могут пользоваться студенты, исследователи и вообще любой желающий для некоммерческих или открытых проектов³¹.

Бесплатный план **не требует привязки карты** и предоставляет полный функционал, но все проекты при этом публикуются в открытом доступе на Roboflow Universe (т.е. ваши датасеты и модели становятся общедоступными) ³¹. Ограничения бесплатного тарифа достаточно щадящие – например, допускается загрузить до **10 000 исходных изображений** ³², а аугментированных версий можно генерировать гораздо больше. Если же требуется хранить данные приватно (что актуально для коммерческих или чувствительных данных, например, видеозаписи с аэродрома), или нужны расширенные возможности/ресурсы, Roboflow предлагает несколько платных планов для бизнеса (Basic, Growth, Enterprise). В них обеспечивается приватность данных, больше квоты по объёму, приоритетные вычислительные ресурсы и некоторые **премиум-функции** (например, большой лимит авторазметки, мониторинг развернутых моделей, поддержка SLA и т.д.). Можно начать с бесплатной **двухнедельной пробной версии** платного тарифа, чтобы оценить преимущества ³³. Отдельно стоит отметить, что Roboflow берёт на себя и лицензионные вопросы используемых моделей: многие распространённые модели типа YOLO имеют ограничения на коммерческое использование, но при работе через Roboflow **эти лицензии покрываются** подпиской ³⁴.

Разумеется, Roboflow – не единственный вариант. Существует ряд аналогичных платформ и инструментов, которые могут вписаться в нужды автоматизации CV-конвейера:

- **SuperAnnotate** – платформа для разметки и управления данными, с возможностями обучения моделей и активного обучения. Предлагает богатый интерфейс для аннотирования (Bounding box, сегментация, полярные координаты для ротированных объектов и т.п.) и интеграцию с пайплайнами. Подходит для команд, но значительная часть функций доступна по коммерческой подписке ³⁵.
- **V7 Darwin** – мощный инструмент для разметки (особенно сегментации) с поддержкой авто-аннотаций на основе нейросетей. Имеет встроенный тренер моделей и удобный веб-интерфейс для экспериментов.
- **Encord** – платформа enterprise-уровня, ориентирована на управление большими датасетами и активное обучение. Предлагает инструменты для label quality, curation и обучения моделей, позиционируется как альтернатива Roboflow для крупных проектов ³⁶.
- **Hasty.ai** – платформа, фокусирующаяся на ускоренной разметке с помощью нейросетей. Позволяет в процессе разметки обучать внутренние модели, которые начинают ассистировать аннотаторам. Также поддерживает цикл тренировки моделей.
- **Labelbox, Dataloop** и др. – похожие экосистемы, предоставляющие интерфейсы для аннотации и управления данными, с определёнными возможностями интеграции в ваши ML пайплайны. Они больше ориентированы на этап разметки и менеджмента данных, тогда как обучение/деплой часто остаётся за пользователем. Тем не менее, они могут быть полезны, если основной упор проекта – в сборе большого количества размеченных данных.
- **Cloud AutoML сервисы** (Google Cloud Vision AutoML, Azure Custom Vision, AWS Rekognition Custom Labels) – дают возможность обучить модель классификации или детекции, загрузив данные через веб-интерфейс. Эти решения особенно удобны для быстрого прототипирования *простых случаев* (например, классификатор «есть повреждение на детали или нет», или детектор определённого объекта). Их плюс – полная абстракция от модели (вы вообще не выбираете архитектуру, всё делает провайдер) и лёгкий хостинг в облаке. Минус – как правило, ограниченность типов задач: **сегментацию или позу** через них не обучить, только классификацию и Bounding Box детекцию. К тому же, стоимость при больших объёмах данных может значительно превосходить использование специализированных CV-платформ или собственного GPU.

- **Open-source инструменты:** Если у команды есть ресурсы разработчиков, можно собрать конвейер и из отдельных открытых компонентов. Например, использовать **LabelStudio** или **CVAT** для веб-разметки (self-hosted, бесплатно), затем обучать модели с помощью **Ultralytics YOLO** (pip-библиотека) или **Detectron2/MMDetection** (для DETR-подобных), а для деплоя использовать **ONNX Runtime** или **Nvidia DeepStream** на своём оборудовании. Такой подход даёт максимальный контроль и отсутствие лицензионных платежей, но требует больше времени на интеграцию всех частей. Некоторые инструменты облегчают и эту задачу: например, **Ultralytics HUB** – это попытка создать упрощённый аналог Roboflow (загрузка данных, тренировка YOLO, мониторинг) в облаке от авторов YOLOv5/v8. Также проекты вроде **Matroid** предлагают полностью готовые аплайнсы для промышленного видеомониторинга: пользователь выбирает интересующие объекты/события, и система сама разворачивает соответствующие детекторы и выдаёт оповещения.

Важно при выборе инструмента оценить, что для вас приоритетнее – **минимум собственного кода** и быстрота развёртывания, либо полная автономность и гибкость настройки. В условиях небольшого пилотного проекта (например, создание системы видеомониторинга перрона на одном участке аэропорта) платформы типа Roboflow дадут выигрыш по скорости: за считанные недели можно получить работающий прототип, подкреплённый мощными моделями. В долгосрочной перспективе (масштабирование на весь аэропорт, интеграция в существующие системы безопасности) может оказаться выгоднее переход на self-hosted решения, чтобы снизить зависимость от сторонних сервисов и гибче адаптировать под себя (тем более, что Roboflow и аналоги позволяют выгрузить модели, не замыкая вас навечно на своём сервисе).

Real-time внедрение и требования к оборудованию

Часть рассматриваемых кейсов подразумевает **обработку данных в реальном времени** – например, анализ видеопотока с нескольких статических камер вокруг самолёта или трансляция с нагрудных камер техников в ходе обслуживания. Реализуя такие системы, нужно учитывать вычислительную нагрузку и подбирать оборудование, способное обеспечить нужную частоту кадров с минимальной задержкой. Рассмотрим несколько примерных сценариев и соответствующие требования:

- **Одиночная камера, умеренное разрешение (Full HD), нерегламентированный FPS.** Допустим, камера фиксирует процесс подготовки самолёта к вылету и раз в секунду кадр анализируется на наличие ключевых событий (подъехал ли трап, убраны ли колодки, подключён ли кондиционер). Здесь нет жёсткого требования к 30 кадрам/сек; анализ 1–5 FPS вполне приемлем. В таком случае даже одна **производительная видеокарта уровня NVIDIA RTX 4080** способна справиться: на ней параллельно могут работать и YOLOv8 детектор, и сегментатор SAM 3 для уточнения, с суммарной задержкой считанные сотни миллисекунд на кадр. Как отмечалось, SAM 3 **вполне работает на RTX 4080**, укладываясь по памяти и давая приемлемый throughput ⁹. Узким местом может оказаться скорее скорость доступа к диску или CPU overhead при обработке большого видеопотока, но при одиночной камере это не критично. По сути, **4080 с 16 ГБ** – это минимальный комфортный уровень GPU для работы с передовыми моделями (даже с запасом) ⁹, позволяющий в реальном времени или околореальном времени обрабатывать поток видео небольшого разрешения offline.
- **Многокамерная система или высокое разрешение (4K) с требованием к высокой частоте кадров.** Например, нужно одновременно отслеживать ситуацию вокруг нескольких самолётов (скажем, 5 камер 4K по 15 FPS). Поток данных огромен, и даже самые оптимизированные модели не смогут обрабатывать все кадры последовательно на одном

GPU. Решения здесь могут быть такими: а) **масштабирование по оборудованию** – задействовать несколько GPU (кластер) либо использовать серверные ускорители вроде **NVIDIA H100/H200**. Последний, как мы упоминали, способен выдавать ~30 мс на кадр для SAM 3 ⁹, а детекторы вроде RF-DETR или YOLO на такой карте могут работать ещё быстрее (сотни FPS суммарно). б) **Оптимизация и разреженная обработка** – например, запускать тяжёлую модель не на каждом кадре, а раз в N секунд, или по событию. Можно использовать лёгкий детектор для постоянного мониторинга, а тяжёлую сегментацию включать только когда выявлено что-то подозрительное. Кроме того, помогает **ограничение числа объектов**: как отмечают разработчики SAM 3, время обработки растёт примерно линейно с числом одновременно отслеживаемых объектов ¹⁰. Поэтому для строжайших realtime-сценариев иногда **ограничивают классы интереса** (например, только люди и транспорт) или количество одновременно сопровождаемых экземпляров (топ-5 крупнейших объектов и т.д.), чтобы гарантировать стабильный FPS.

- **Edge-аналитика (на борту устройства или переносном модуле)**. Сюда относятся случаи, когда поток с камеры должен анализироваться прямо **на месте**, без высокопроизводительного сервера. Пример: нагрудная камера сотрудника, подключённая к мобильному вычислителю (типа Jetson Orin Nano, планшет с TPU, и т.п.), которая в режиме реального времени предупреждает о нарушении (вибро-сигналом, если человек заходит в опасную зону). Здесь использование SAM 3 или других “больших” моделей исключено – требуется очень компактная нейросеть и аппаратное ускорение. В подобных сценариях обычно применяют **модели Nano-уровня**: например, **YOLOv8n** или специально оптимизированные модели (MobileNet-SSD, PP-YOLO Tiny). Они могут работать на одноплатных компьютерах. Так, опыт показывает, что на NVIDIA Jetson Orin NX (15 Вт) модель YOLOv8n способна обрабатывать 30+ FPS на 640p видео. Для задач классификации можно применять **ускорители типа Google Coral TPU**, которые эффективно выполняют небольшие модели. Также существуют специализированные библиотеки (NVIDIA TensorRT, OpenVINO) для сжатия и ускорения инференса – их стоит использовать на этапе деплоя. Однако разработка под edge часто требует тщательного профилирования: может оказаться, что лучше пожертвовать некоторой точностью (обучив модель поменьше размером), но гарантировать её работу в реальном времени на доступном устройстве.

Пример оборудования и производительности:

- **Jetson AGX Orin 32 ГБ**: способен выполнять модели ~YOLOv5m (Средней размерности YOLO) в районе 15 FPS на 1080p, или YOLOv8n – 100+ FPS. Подходит для установки прямо на спецтранспорт или в носимом формате (рюкзак с модулем).
- **NVIDIA L4 или T4 (PCIe ускорители)**: один такой ускоритель (PCIe-карта, 16 ГБ) может в реальном времени тянуть 2–3 потока Full HD с YOLOv8s и простой сегментацией. Это вариант для развертывания в существующем сервере.
- **NVIDIA H100/H200**: датацентрические карты, способные обрабатывать десятки потоков 4K. Например, разница между H100 и H200 – около 30% прироста; H200 при batch-инференсе ~30 изображений параллельно выдаёт невероятный throughput ³⁷. Такие GPU оправданы, если нужно покрыть сразу весь аэропорт одной серверной стойкой, но это очень дорогое решение. Для начальных проектов чаще пойдут **серии 30xx/40xx или специализированные edge-устройства**.

Наконец, стоит помнить: real-time – понятие растяжимое. Если речь о **safety-критичных** вещах (например, автопилот рулёжки самолёта по аэродрому), то требования жёстче (миллисекунды задержки). Если же цель – **отчет/фиксация нарушений** (скажем, записать факт опоздания спецмашины к самолёту), то обработка с задержкой в секунды или пост-аналитика после события тоже приемлемы. В нашем случае многие задачи (контроль графика обслуживания,

документирование) вполне допускают режим *near-real-time* или *offline*. Поэтому при проектировании конвейера всегда можно разделить поток на две части: **оперативный модуль** (легковесный детектор на edge, выдающий сигналы тревоги/события) и **тяжёлый аналитический модуль** (работает в облаке или сервере, подробно анализирует собранные данные, возможно, задействуя SAM3, 3D-модели и прочие ресурсоёмкие алгоритмы). Такой разделённый подход обеспечивает баланс между скоростью реакции и глубиной анализа.

Заключение

Подводя итог, для автоматизации мониторинга технологических процессов в гражданской авиации сегодня доступны беспрецедентно мощные и удобные средства компьютерного зрения. Конвейер построения CV-моделей включает: **сбор разнообразных данных** (статичные камеры, носимые камеры – обязательно учитывая различия в ракурсах и динамике), **интеллектуальную разметку** (с помощью моделей вроде SAM 3 для ускорения и повышения качества аннотаций), **обучение современных моделей** (как классических YOLO, так и трансформеров типа RF-DETR, которые всё чаще занимают лидирующие позиции по сочетанию скорости и точности ²¹ ²⁴), а также **развёртывание с учётом требований реального времени** на соответствующем оборудовании. Особый упор следует делать на **гибридные решения**: совмещение разных типов моделей (детекция + сегментация + поза + классификация) даёт более полную картину происходящего, а продуманное распределение задач между edge-устройствами и серверными мощностями позволяет достичь и скорости, и точности. Инструменты вроде Roboflow значительно снижают порог внедрения таких технологий, позволяя сфокусироваться на бизнес-логике (какие события отслеживать и как реагировать) вместо рутинной технической работы. При этом наличие альтернатив и открытых решений означает, что выбранную в пилоте технологию всегда можно перенести “в дом” или масштабировать оптимальным образом. Грамотно выстроенный CV-конвейер станет надёжным помощником в обеспечении безопасности и соблюдении регламентов в аэропорту – от контроля за каждым винтиком до мониторинга слаженной работы всей наземной службы.

Источники: Использованы открытые материалы Meta AI и Roboflow по модели Segment Anything 3 ⁶ ¹⁰ ⁹, обзоры Roboflow по модели RF-DETR ²¹ ²⁰ и гибридным архитектурам детекторов ¹⁶ ¹⁵, руководства Roboflow по развёртыванию CV-моделей и поддерживаемым возможностям (включая позу-детекцию ²⁶). Также учтены данные о производительности GPU (H200, RTX 4080) при инференсе современных моделей ⁹ и сведения о планах доступа Roboflow ³³ ³¹ для оценки целесообразности использования платформы в корпоративных проектах.

¹ ² New Segment Anything Models Make it Easier to Detect Objects and Create 3D Reconstructions
<https://about.fb.com/news/2025/11/new-sam-models-detect-objects-create-3d-reconstructions/>

³ ⁴ ⁵ ⁶ ⁷ ⁸ ⁹ ¹⁰ ¹¹ ¹² ³⁷ SAM 3: Segment Anything with Concepts
<https://blog.roboflow.com/what-is-sam3/>

¹³ ¹⁴ YOLO Object Detection Explained: Real-Time Vision Tasks
<https://blog.roboflow.com/yolo-object-detection/>

¹⁵ ¹⁶ ¹⁷ ¹⁸ ¹⁹ ²⁰ ²¹ ²² ²³ ²⁴ ²⁸ ²⁹ RF-DETR: A SOTA Real-Time Object Detection Model
<https://blog.roboflow.com/rf-detr/>

²⁵ Best Pose Estimation Models & How to Deploy Them
<https://blog.roboflow.com/best-pose-estimation-models/>

26 27 **Launch: Label, Train, Deploy Support for Keypoint Detection Models in Roboflow**
<https://blog.roboflow.com/keypoint-detection-on-roboflow/>

30 **How to Use the Yolo Real-Time Object Detection Detection API**
<https://universe.roboflow.com/my-workspace-zdfxw/yolo-real-time-object-detection/model/4>

31 33 34 **Roboflow Pricing and Plans**
<https://roboflow.com/pricing>

32 **What are the image limits for our free plan? - Roboflow's Forum**
<https://discuss.roboflow.com/t/what-are-the-image-limits-for-our-free-plan/434>

35 36 **Top 10 Roboflow Alternatives & Competitors in 2025 | G2**
<https://www.g2.com/products/roboflow/competitors/alternatives>